電腦時代 COMPUTING AGE 8

微型電腦用家雜誌 HK$8.00

PROMAS 305 3吋超小型磁碟機

小巧強勁！

- 143K儲存量，完全適用於任何APPLE* 操作系統(OPERATING SYSTEM)，並可與傳統5吋磁碟機及**DISK DRIVE CARD**共用。
- 體積爲普通5.25吋磁碟機之三份一，易於放置。
- 全部日本製造及裝嵌。
- 眞正可作兩面使用。

* TRADE MARK OF APPLE COMPUTER INC

PROMAS ©

總代理
VINCENT COMPUTER CENTRE LTD.
雲生電腦有限公司
九龍尖沙咀柯士甸路22-26號好兆年行6字樓 電話：3-7211288-0 電訊：HX 38316 VCC HK

TAXAN RGB「幻彩系列」

彩色電腦螢幕
色彩艷麗完美無瑕

- **能顯示任何顏色組合**
  率先採用「線性影像放大系統」
- **絕不干擾磁碟及其他電器**
  設計符合多項美國嚴格標準

- **適用於各款電腦**
  備有多種附件，適用於下列電腦：
  RGB（紅綠藍）彩色介面
  APPLE II[1] APPLE IIe[1]（連80字顯示）
  介面接駁線
  APPLE III[1] IBM PC[2] BBC[3] NEC PC[4]
  FUJITSU FM-7[5] COMMODORE[6]
  歡迎訂造特別介面接駁線

1. APPLE IS A TRADE MARK OF APPLE COMPUTER INC.
2. IBM IS A TRADE MARK OF INTERNATIONAL BUSINESS MACHINE CORP.
3. BBC IS A TRADE MARK OF ACORN INTERNATIONAL LTD.
4. NEC PC IS A TRADE MARK OF NEC CORP.
5. FUJITSU FM-7 IS A TRADE MARK OF FUJITSU LTD.
6. COMMODORE IS A TRADE MARK OF COMMODORE ELECTRONICS LTD.

享有代理商保證
才是專業性採用

總代理
VINCENT COMPUTER CENTRE LTD.
(A Subsidiary of Vincent Group)
雲生電腦有限公司（雲生集團附屬機構）
九龍尖沙咀柯士甸路22 [illegible] 608室 電話：[illegible] 電傳：HX3031C XCC HK

LEE'S ELECTRONIC CO.

灣仔總店

Z-80 PLUS

- Totally compatible with all CP/M software, there are no exceptions!
- Fully compatible with Microsoft disks, most other Z-80 card are not.
- Runs on any Apple IIe, II or II+ with at least 48K memory, works with any 80 column card but does not require one.

NEW JOY-PORT III

特別設計全新綫路，絕對可玩全部Game，可以選擇AUTO FIRE，配合QUICK SHOT JOYSTICK，

注意

QUICK SHOT JOYSTICK 如果唔加JOYPORT CARD 唔可以玩LADY TUT，或者將來新Game

SPEECH CARD II

中文，英文說話卡，更可幫你讀出CATALOG，DOS COMMAND，BASIC PROGRAM，外接式高品質小型喇叭箱，聲音響亮清晰Software+Manual

香港灣仔道128-150號海運商場1樓
1-2室　電話：5-8921792
Shop No. 1-2 Ocean Shopping Centre, 1/F., 128-150 Wanchai Rd., Hong Kong.
TEL: 5-8921792

北角分店

ATARI®

600XL 64K RAM EXPANSION

- Upgrade 600XL to 800XL
- 1/2 year warantee
- unbelievable low price

COMPUTER DATA RECORDER FOR APPLE COMPUTER

電腦專用錄音機獨有SAVE和LOAD鑑聽和指示 AUTO STOP帶長COUNTER保用三個月

*"香港區總代理，歡迎批發"

SOFTSWITCH

80 Column Card 新配搭，40—80Column自動上落 絕對可行 PFS Program

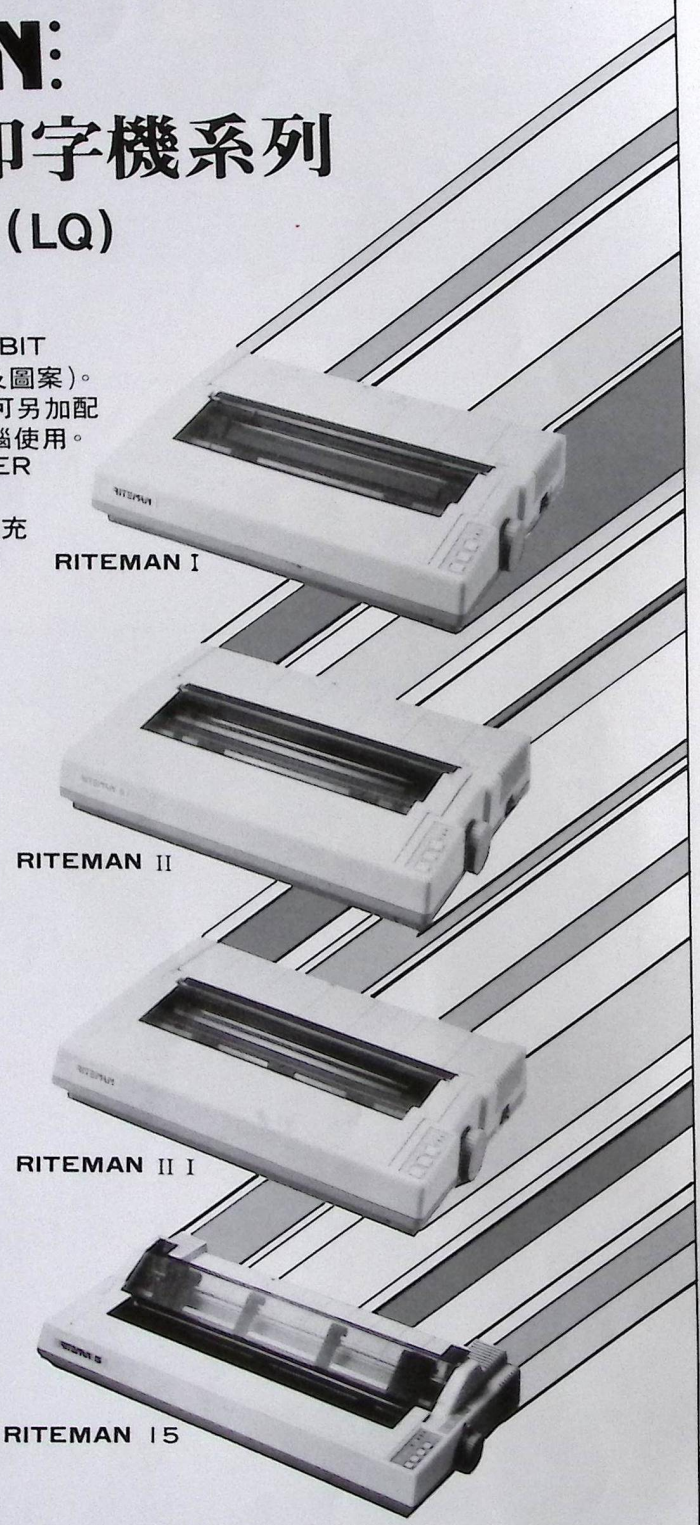
精工修理

Apple II® 主機　Apple II® 磁碟機

香港北角英皇道255號國都商場3樓
213室　電話：5-718392
Shop No. 213 Olympia Shopping Centre, 2/F., 255 King's Rd., North Point, H.K.
TEL: 5-718392

全新 :RITEMAN:
「威猛牌」超級印字機系列
新貨、新價、新功能(LQ)

- 超薄機身，體積爲普通印字機一半。
- 多種印字功能，可印多國文字，更可作BIT IMAGE GRAPHIC DUMP(中國文字及圖案)。
- 標準CENTRONICS平行式介面接口，可另加配串行式RS 232介面接口，適合任何電腦使用。
- 可用軟件及硬件選擇「高質素」(LETTER QUALITY)字體，適用於文字處理。
- 採用新設計之盒式絲質墨帶，用戶可補充油墨重覆使用，操作簡易、墨帶壽命高達一百萬字，減少更換墨帶次數，不斷節省金錢，物超所值。
- 系列最完備，提供更多選擇。



RITEMAN I
- 印字速度每秒120字
- 可印4國文字

RITEMAN II
- 印字速度每秒160字
- 可印8國文字

RITEMAN III
- 印字速度每秒140字
- 可印8國文字
- 完全適用於IBM P/C* 及任何電腦

RITEMAN 15
- 可用15吋紙
- 完全適用於IBM P/C* 及任何電腦
- 印字速度每秒160字

*IBM P/C爲萬國商業機器註册商標

總代理：
VINCENT COMPUTER CENTRE LTD.
雲生電腦有限公司
九龍尖沙咀柯士甸路22－26號好兆年行6字樓 電話：3-7211288-0 電訊：HX 38316 VCC HK

香港電腦研習中心

Hong Kong Centre of Computer Studies

地址： 九龍旺角太子道141號長榮商業大廈506室

電話： 3-804040

本校特點：

一、師資優良：
全部講師爲香港大學畢業或以上程度。持有電腦或有關科目學士或碩士學位，並富有教學經驗。

二、充足實習時間：
置有十餘部電腦，供學員實習。星期一至六，上午十時至晚上八時，學員可隨時回校實習，不另收費。

三、備有教學軟件，學員可利用課餘時間，隨電腦學習。

四、講義由經驗講師精心編寫，深入淺出。

五、交通便利：
本中心位於旺角長榮商業大廈，即大大公司斜對面，地鐵太子站出口，恒生銀行樓上。歡迎參觀及查詢。（星期一至六，上午十時至晚上九時半）。

Advanced BASIC Programming

包括 programming and debugging techniques; CALL, PEEK, POKE; shape table, animated graphics music ; DOS; CP/M; data file programming; sorting and searching 等。全期學費連講義及實習共320元。

Introduction to Computer Programming

以教授BASIC電腦語言爲主，並包括電腦知識及磁碟操作系統（DOS），適合初學。全期學費連講義及實習共300元。

Word Processing with WordStar

以電腦代替打字機，處理文字。除Word Star外，兼授Mailmerge及SpellStar， 適用於任何能行CP/M之電腦。學費300元。

Certificate Course in Computer Business Application

專爲工商界初學電腦人士而設，適合各類型電腦。

1. Electronic Spreadsheet
2. Database Management
3. Word Processing

爲期三月，共分兩期，學費每期350元。

Fundamentals of dBASE II

用dBASE II處理資料兼授CP/M, 學費300元。

Programming in dBASE II

用dBASE II 設計商用程式兼授QUICKCODE及 dUTIL，學費360元。

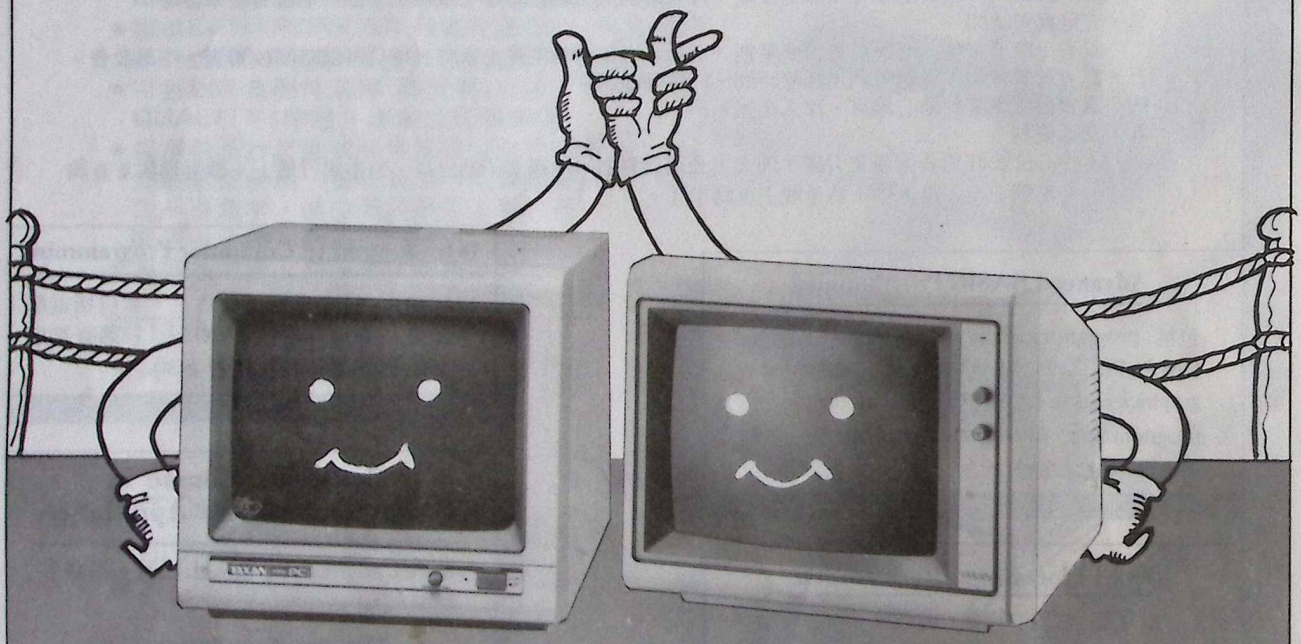
Apple Machine and Assembly Language

包括6502機械語言及LISA Assembler，全期學費連講義及實習共360元。

迎接電腦世紀的來臨

學習電腦語言與電腦溝通

TAXAN® 電腦螢幕
與IBM PC*私人電腦
最佳配搭 出類拔萃



TAXAN「KX 1212/1213」單色電腦螢幕及「VISION-PC」RGB彩色螢幕特別爲 IBM PC* 及同類型電腦而設計，功能及質素均可媲美IBM*原廠產品。

優點特多：

- 特高解像度，視頻特寬。
- 新設計方形平面黑底不反光顯像管，加強畫面反差度，久看不倦。
- KX 單色螢幕備有綠色「長殘影」及琥珀色（橙色）型號，可供選擇。

請駕臨各大分銷商參觀比較。

* IBM及IBM PC爲萬國商業機器註冊商標

享有雲生電腦保証
才是專業性的保用

總代理：
VINCENT COMPUTER CENTRE LTD.
雲生電腦有限公司

九龍尖沙咀柯士甸路22-26號好兆年行6字樓　電話：3-7211288-0　電訊：HX 38316 VCC HK

★ 學習尖端科技

★ 邁向成功坦途

現代電腦系統訓練中心

MODERN COMPUTER SYSTEM TRAINING CENTRE

——暑期電腦課程招生——

COURSE 課程表	上課編制	時數	學費
電腦語言課程			
(1) BASIC PROGRAMMING COURSE	個別或班制	16HRS	$280
(2) ADVANCED BASIC COURSE	個別或班制	15HRS	$300
(3) ASSEMBLER LANGUAGE COURSE	個別或班制	16HRS	$350
(4) "C" LANGUAGE COURSE	班制上課	16HRS	$800
(5) 兒童電腦班（LOGO及初級BASIC）	班制上課	10HRS	$200
電腦硬件課程			
(1) 6502電腦自嵌實習班	個別或班制	30HRS	$500
(2) APPLE-II電腦修理班	班制上課	80HRS	$1500
(3) IBM-PC電腦修理班	班制上課	80HRS	$1800
綜合應用課程			
(1) dBASE-II入門及商業應用初級及高級班	個別或班制	12HRS	$300
(2) CP/M操作系統	個別或班制	16HRS	$350
(3) DOS應用及拆碟班	班制上課	20HRS	$550
(4) WORDSTAR TRAINING COURSE	個別或班制	16HRS	$350
(5) PEACHTREE A/C SYSTEM會計系統	個別或班制	16HRS	$800

上述學費概以班制上課爲準，個別教授另議

創辦兩年　電腦碩士任教

本校師資 (1)香港及中文大學電腦系，英美加電腦系畢業生。 (2)英國曼徹斯特大學電腦碩士，英國電腦學會會員。

本中心開放時間：10：00A.M—10：00P.M

地址：九龍深水埗福華街156號黃金商場地庫20號室　電話：3-601531,3-7289080

分址：荃灣大壩街67號閣樓　旺角太子道141號長榮商業大厦303A號（彌敦道口）

0-429147　3-869085

中華基督教青年會 黃傳經中心

香港基督教女青年會 總會所

IBM電腦課程

IBM COMPUTER COURSE

I. INTRODUCTION TO COMPUTER 電腦入門

1. A Guide to Computer
2. Historical Development
3. What is MS-DOS Operational System
4. Introduction to 8088 CPU
5. Introduction to IBM—PC Structure

II. BASIC PROGRAMMING IBM程序編寫

1. The Knowledge of BASIC Language
2. Program Flowchart
3. Program Concept

III. BUSINESS PROGRAM DESIGN TECHNIQUE
商用程式設計技巧

1. Selecting and Sorting
2. Tax and Invoice Program Design
3. Application Program Development

IV. IBM COMPUTER OPERATION IBM電腦操作

1. How to Format a Diskette
2. How to Operate a Disk Drive
3. Searching File
4. Read and Write File

Remark: Tuition fee for the whole course is $380, including notes lectures and practices.

全期兩個月，費用 $380 包括理論及實習

歡迎查詢

Y.M.C.A 青年會黃傳經中心
香港西灣河街 46-56 號西灣河大廈二樓
5-671571

香港基督教女青年會
中環麥當奴道1號
5-223101 EXT 10

中華基督教青年會 黃傳經中心

香港基督教女青年會 總會所

IBM 中級電腦課程

IBM ADVANCED BASIC COURSE

I. IBM MS-DOS OPERATION TRAINING
(IBM MS-DOS 操作訓練)

1. What is MS-DOS operational system
2. Introduction to 8088 CPU
3. Introduction to IBM—PC structure

II. ADVANCED PROGRAMMING TECHNOLOGY
(高級電腦程序編寫技巧)

1. Bubble Sorting
2. Shell or Quick Sort
3. Merging

III. FILE OPERATION (檔案處理)

1. Writing and Reading File
2. Searching File
3. Program File Handling
4. Sequential File Structure
5. File Editing on Sequential File
6. Random File Structure
7. File Editing on Random File

IV. PROJECT DESIGN FOR BUSINESS APPLICATION.
(IBM 商用程式設計實習)

Remark: Fee for the whole course is $380 including practice.

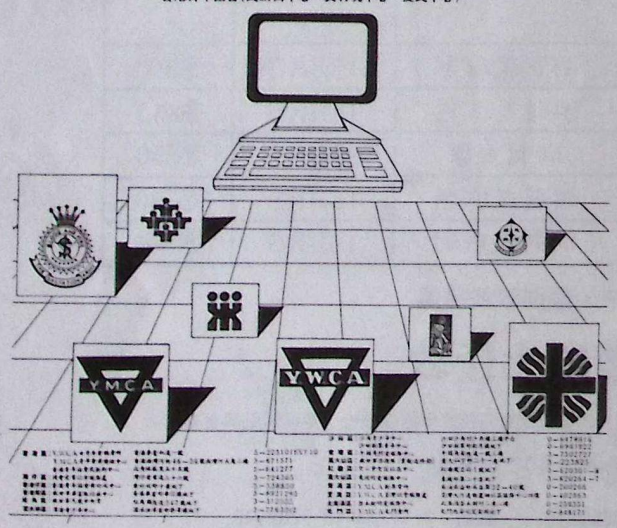
全期共兩個月，費用 $380 包括理論及實習
歡迎曾學初級課程者繼續進修

Y.M.C.A 青年會黃傳經中心
香港西灣河街 46-56 號西灣河大廈二樓
5-671571

香港基督教女青年會
中環麥當奴道1號
5-223101 EXT 10

電腦證書課程

• 香港中華基督教青年會(黃傳經中心、新界會所、必列者士街會所、屯門會所)•
• 香港基督教女青年會(香港總會所、柏顧斯會所、荃灣會所)•
• 救世軍中央堂"業餘進修部"、華富隊社區中心、啟德隊"業餘進修部"•
• 東頭明愛服務中心、牛頭角明愛服務中心、梨木樹明愛服務中心•
• 浸會愛羣社會服務處•聖公會聖匠社區中心•沙角青少年中心•
• 香港青年協會(民新街中心、黃竹坑中心、愛民中心)•



中華基督教青年會

香港基督教女青年會

升學及就業
電腦專業訓練課程

(全日制)

課程特點：本課程爲準備開學者可獲得各種常用電腦語言及其程序編寫之知識，爲從事電腦行業者，提供商業電腦軟件之應用，電腦打字，電腦會計，人際關係等實用科目。

程度：F.5 或以上程度均可報名。(全日制)

課程編訂：課程之編訂及按工商界需求，俾能學以致用。學員完成課程，可自行應用商業及私人電腦，成績良好可獲證書，證書由 (Y.M.C.A.)／(Y.W.C.A.)頒發。

費用：每月三百八十元 (全期九個月)，首期繳交二個月，後七期每月繳交 $380。

歡迎查詢

香港基督教女青年會 Y.W.C.A.
麥當奴道 1 號
5—223101 EXT.10

中華基督教青年會 Y.M.C.A.
窩打老道 23 號
3—319111

Z80咭淺談

陸生

APPLE Ⅱ 的操作系統，除了在6502微處理器下的DOS 3.3之外，還有後期發展完成的CP／M操作系統。但CP／M 的設計是基於微處理器8080／E—80的，而且全球性已有了一個標準，故此APPLE Ⅱ 要執行CP／M下的程式，便要加上一張 Z — 80 咭。Z—80咭與CP／M的相關用途及操作，下文將作略述。

(1)Z－80咭的裝置

Z— 80 咭又稱爲軟性卡（ SOFT CARD），它與其他的週邊咭一樣，是放在APPLE Ⅱ 的外展插座中。基本上，Z — 80 咭應該放在4號插座（ SLOT 4） 上，但也可以放在7號插座上，當Z — 80 咭裝置了之後，APPLE Ⅱ便可以在兩個操作系統中，利用命令， 任選其一。當APPLE Ⅱ 在6502的環境操作時，Z — 80 咭絕不會影响它正常的操作，當APPLE Ⅱ在Z—80下工作時，6502亦不會對它有所影响，而且APPLE Ⅱ 還可以運行CP／M 操作系統及MBASIC，GBASIC等。Z—80咭的裝置非常簡易，毋須任何的硬件或軟件修改，而Z—80咭之所以得名，因爲在咭之上確有一隻Z—80A微處理器，而令APPLE Ⅱ可以執行基於Z—80微處理器而編寫的程式。

(2)CP/M操作系統

撇開Z—80咭不說，CP／M操作系統可以說是最重要的一環，它可以令到衆多的Z—80程式在APPLE Ⅱ 上執行。在APPLE Ⅱ 的CP／M版本爲 2．0，而它又可分爲 44K，56K， 及 60K 三種系統。CP／M 操作系統包括了很多輔助程式，而它們所能提供的功能，包括了資料在磁碟上存取，傳送、編印、翻譯等等。CP／M 操作系統被廣泛地利用於 8080及Z—80家族的微處理器上，相對地，更多及更廣的應用軟件被開發及寫成，爲用家提供了很大的方便及軟件使用的靈活性。一般來說，APPLE 的用家可以與其他的CP／M操作系統下的程式互相交流使用，但礙於APPLE Ⅱ 的CP／M磁碟有其獨特的結構，要把其他電腦的程式轉送至APPLE Ⅱ 中，便需要經由一個程序來做，這個程序就叫做〔DOWN LOAD〕。CP／M與DOS 的分別就是它有一個較快的資料傳送法，檔案轉移，傳送，容易的機器語言介面及多重檔案找尋（WILD—CARD FILE NAMING CONVENTION）。而在CP／M上的MBASIC，又有著許多新的命令，如"PRINT USING"，"CALL"，"WHILE／WEND"，"CHAIN"，"COMMON"等等。

(3)重要的專用名詞

在未更深入地研究Z—80咭之前，我們必須對以下的專用名詞作一認識：

1、44K系統（44 K SYSTEM）

這是指基本的APPLE Ⅱ（48K）而言，在CP／M系統時，APPLE Ⅱ 只有44K的地址可以作爲資料的存取，餘下的4K，是用來作爲APPLE Ⅱ 的螢幕及CP／M磁碟扇區的讀寫程序而設。

2、56K系統（56K SYSTEM）

這是指APPLE Ⅱ 加上了16K RAM咭或語言咭（LANGUAGE CARD）而言。在原本的64K系統中，除了4K要用來處理APPLE Ⅱ 的螢幕畫面及扇區讀寫程序之外，另外因爲在16K RAM咭上只有12K的地址可被存取，故此有4K BYTE 的記憶容量是浪費了的。

3、A：—F：

CP／M操作系統對其磁碟機的稱呼，在6號插座上第一個的磁碟機爲A：，第二個爲B：，在5號插座上第一個的磁碟機爲C：，在第4號插座上第一個的磁碟機爲E：，餘此類推。

4、終端機（EXTERNAL TERMINAL）

這裡是指80字行咭而言，在CP／M操作系統中，它自動把3號插座上的80字行咭看成一個終端機，如沒有80字行咭，則訊號會經由一般的視頻輸出端作資料的傳送，這時便只得24×80的畫面規格。

(4)正確的介面咭/插座編法

在Z—80咭未裝置之前，用家必須對APPLE Ⅱ 上的外展插座及其相關的介面咭有一定的認識，因爲基本的CP／M操作系統的編寫程序，是根據特定的介面咭／座的安排而寫成，以下爲各介面咭與相應之插座編配表。

插座	用途/介面咭
0	語言咭16K RAM 咭，16K ROM咭。後者不爲CP／M所採用
1	印字機介面咭
2	一般性的資料傳輸埠，如MODEM，紙帶機讀寫器
3	80字行咭
4	Z—80咭，磁碟機介面咭（E：及F：）
5	磁碟機介面咭（C：及D：）
6	磁碟機介面咭（A：及B：）
7	沒有特定用途Z—80咭亦可放在這裡

(5)Z80與6502的地址編譯

在6502上記憶區地址編譯法與Z—80的是不一樣的，要達到相互地址的編譯任務便落在Z—80咭上的一組硬件綫路來担任。以下爲兩個微處理器（6502／Z—80）相同的記憶區在地址間的呼叫法。

全球最佳家庭電腦
隆重在港推出
COMMODORE 64

程式 600 個
免費贈送

八三年全世界銷量冠軍*
八三年全美國銷量冠軍*
歐美七家著名電腦雜誌評選爲「八三最佳家庭電腦」
首家微電腦製造廠年營業額超越十億美元

•論質量•論軟件•論性能•論價格
COMMODORE 64 皆傲視同羣

- 質量：本港發售的Commodore 64均在英國製造，質量非常可靠。
- 軟件：Commodore 64擁有數量龐大的軟件。例如教育方面，有超過六百種不同程序，內容包括電腦、英文、法文、數學、地理、歷史、自然科學等，全部免費提供用戶抄錄。Commodore 64還提供大量遊戲和商用軟件，而且種類日益增多。
- 性能：Commodore 64採用6510 CPU, 64KB RAM, 16種顏色，並可選用近十種不同電腦語言和CP/M。
- 價格：只賣HK$2388，您便可立刻在家中享用Commodore 64帶給您的各種好處。

總代理：鷹達企業有限公司 灣仔洛克道200號東新商業中心17樓 電話：5-8934447

* 售價在US$500以下的微型電腦

Z80 地址	6502 地址
0000H-0FFFH	$1000-$1FFF
0000H-0FFFH	$1000-$1FFF
1000H-1FFFH	$2000-$2FFF
2000H-2FFFH	$3000-$3FFF
3000H-3FFFH	$4000-$4FFF
4000H-4FFFH	$5000-$5FFF
5000H-5FFFH	$6000-$6FFF
6000H-6FFFH	$7000-$7FFF
7000H-7FFFH	$8000-$8FFF
8000H-8FFFH	$9000-$9FFF
9000H-9FFFH	$A000-$AFFF
A000H-AFFFH	$B000-$BFFF
B000H-BFFFH	$C000-$CFFF
C000H-CFFFH	$D000-$DFFF
D000H-DFFFH	$E000-$EFFF
E000H-EFFFH	$F000-$FFFF
F000H-FFFFH	$0000-$0FFF

(6)軟件支緩

Z—80咭可以執行全部8080或Z—80微處理器的指令，順理成章也可以執行基於這些微處理器所寫成的程式，除此之外，還有一些CP／M的實用程式（UTILITY），如ED—傳統的單行文字處理器，ASM，—8080 的澠編器，DDT—8080 編澠語言的偵錯器等，這些實用程式皆可被利用去編寫及偵錯 8080 的程式， CP／M在 APPLE Ⅱ上的記憶區地址用途可見表A。

(7)硬件的特性

在Z—80咭上的Z—80A微處理器，與APPLE 主機板上的時鐘訊號是同步而且被鎖定在2.041兆赫的有效時鐘頻率上，時鐘訊號之產生，全繫於在介面咭上的U4，U1及U9，而晶體管Q1及其他組結則構成了一個類比緩衝區（ANALOG BUFFER）而致能達到高速的轉換開關。

當APPLEⅡ 的電源開啓後，由APPLE而來的重置訊號就令到Z—80咭進入停頓的狀態，接着重置的訊號與主機板上的時鐘訊號相同步，這個作用，就是要令到寫的操作（WRITE OPERATION）不被干擾，這時Z—80便進入等候的狀態，直至有一個寫的訊號去觸發在$C400—$C4FF的記憶地址區，Z—80便被觸發，在Z—80咭上的（E）便亮着，如果再接到另一個相關地址的記憶周期，Z—80咭便從等候狀態釋放出來，但如果Z—80咭對自己的相關記憶位置發出一個寫的訊號時，它便會再度放在等候的狀態中。以下的簡表，爲控制Z—80咭與其插座的地址區。

插座	地址區
1	$C100—$C1FF
2	$C200—$C2FF
3	$C300—$C3FF
4	$C400—$C4FF
5	$C500—$C5FF
6	$C600—$C6FF
7	$C700—$C7FF

(8)咭上的微型開關

在Z—80咭的微型開關，不爲人所認識，但這個開關所負起的任務却是最廣泛的，無論在地址傳輸綫（ADDRES—S BUS）的介面作用，直接儲存器存取雛菊鏈（DMA DAISY CHAIN）及間斷要求訊號（INTERRUPT REQUEST SIGNAL）等等都由這個小小的開關所控制。它們的功能簡述如下：

1、S1—1 接上（ON）

Z—80 咭 的地址綫與 APPLE 的I／O母綫是相銜接的，要做到這個目的，其間就要經過一組的翻譯組件，這個硬體綫路由U7，U8，U11及半個U12所組成。當S

表（A）

Z80 地址	6502 地址	用途
$800—$FFF	0F800—0FFFF	APPLE 的磁碟機及緩衝區
$400—$7FF	0F400—0F7FF	APPLE 的螢幕地址區
$200—$3FF	0F200H—0F3FFH	輸入／輸出對照表（IOB）
$000—$1FF	0F000H—0F1FFH	6502之叠存器（STACK）及零頁區
$C000—$CFFF	0E000H—0EFFFH	APPLE輸入／輸出記憶區對照表
$FFFA—$FFFF	0DFFAH—0DFFFH	6502重定，非間斷干擾及中斷要求指標
$D400—$FFF9	0C400H—0DFF9H	56K系統
$A400—$BFFF	9400H—0AFFH	44K系統

超卓電腦 個別小組 教授中心

本中心特點：

1. 天天開課
2. 直接用電腦教授

效果比班制勝十倍

3. 免費實習
4. 時間任擇

課　　程	個別	小組
Logo (7至12歲)	$400	$300
Computer Operation + Basic Programming	$500	$300
Advanced Basic Programming	$550	$350
Disk Operation System	$550	$350
CP/M & Word Processing	$550	$350
Assembler	$600	$400
dBaseII	$600	$400

以上課程學生一律九折

地址： 九龍尖沙咀漆咸道南3號二樓 （麼地道口近水車屋）

電話： K-669726

訂閱

（請用英文填寫，方便電腦處理）　（只適用於本港訂戶）　信封面請註明「訂閱」

本人欲訂閱

□電腦時代一年（十二期）HK$**96**

□電腦時代＋CompuHouse磁碟一年（十二期）HK$**455**

由第___期開始

姓名：_____

地址：_____

電話：_____

附上劃線支票 No. _____

支票收款人請填寫 "Computing Age Publisher"

請寄九龍中央郵箱71193號

NO. _____

Z 80 咭淺談

1—1 被接通時（ON），這個翻譯組件便失去功用，換言之，此時Z—80的地址傳輸綫便與 APPLE 的 I／O傳輸綫一樣。

2、S 1—1 切斷（OFF）

當S 1—1 被切斷後，翻譯組件便被啓動，它簡單地在每一個地址加上$ 1000 的數值，這便解決了6502 與Z—80 間的地址不協調情況，相對地把 Z — 80的中斷地址及CP／M起始地址改變，使其與6502 的零頁區（PAGE ZERO）分開，有效地令到 CP／M 可以在地址區$C000—$EFFF 作資料的存取。

3、S 1 —2 接上

它可以令到 DMA 的要求爲 Z—80所接受而形成一個中斷要求，這時 DMA 的輸入訊號（27 脚）就會被拉低，直至Z—80完成了最後的一個機器周期時，它便把控制權釋放出來。

4、S 1—2 切斷（OFF）

DMA 的要求不被 Z —80所接受。

5、S 1—3 接上（ON）

Z—80的綫路設計，是可以接受6502 的中斷干擾（INTERRUPT）的，當S 1—3 被接上後，Z—80便可以接受6502 的非間斷性中斷（NON—MASKABLE INT—ERRUPT—NMI）。

6、S 1—3 切斷（OFF）

Z—80不接受 NMI。

7、S 1—4 接上（ON）

有S 1—3 一樣的性能，不過Z—80是接受間斷要求訊號（INTERRUPT REQUEST—IRQ）

8、S 1—4 切斷（OFF）

Z—80不接受 IRQ。 □

兩欄索引顯示

這個短短的程式能將CATALOG 的FILENAME 以左右兩欄方式在螢幕上顯示，換言之，在同一時間最多可列出42 個 FILENAME。對於電腦時代程式磁碟的用家應該非常有用。因爲我們每期的磁碟都有數拾個FILE。當然，本程式功能不單如此，用家可以按單鍵便可RUN 任何的FILE。由於最多可出現42個FILENAME，故筆者還用了ASCII字元來作爲單鍵控制執行字元。

必須留意，要兩欄顯示模式時，FILENAME 必定不能超過十三個字符（正常的是三十個字符），相信讀者們會明白。

各位讀者假如不喜歡程式磁碟上的HELLO程式，可以將此程式作爲H—ELLO程式SAVE 落磁碟上。

本程式在程式磁碟上的FILEN—AME是DOUBLE COLUMN CA—TALOG。□

```
10  REM   **********************
20  REM   * DOUBLE   COLUME  *
25  REM   *     CATALOG      *
30  REM   * BY COMPUHOUSE    *
40  REM   * COPYRIGHT(C)1984 *
60  REM   **********************
100  ONERR  GOTO 2000
110  DEF  FN A(X) =  PEEK (40) +
      PEEK (41) * 256: TEXT : HOME
     :MEM = ( PEEK (978) - ( PEEK
     (978) > 127) * 256) * 256 +
     2947: FOR I = 1 TO 24: READ
     N,D: POKE N + MEM,D: NEXT : GOTO
     1000
400  HTAB H2:BA =  FN A(X):C =  PEEK
     (BA +  PEEK (36)): IF C = 16
     0 THEN  POP : GOTO 1060
401  RETURN
500 T$ = "":BA =  FN A(X): FOR X =
      PEEK (36) TO LE:C =  PEEK (
     BA + X):T$ = T$ +  CHR$ (C):
      NEXT : RETURN
1000  PRINT  CHR$ (4)"CATALOG"
1040  FOR V = 3 TO  PEEK (37) + 1
     : VTAB V:H2 = 1: GOSUB 400: HTAB
     2: PRINT "-";; CHR$ (V + 30)
     ;"-";:H2 = 21: GOSUB 400: HTAB
     22: PRINT "-"; CHR$ (V + 62)
     ;"-";: NEXT V
1060  IF H2 = 21 THEN  PRINT
1080  POKE  - 16368,0: PRINT "ENT
     ER SELECTION#-->";: GET SEL$
     :P =  ASC (SEL$)
1090  IF P - 61 > V OR (P - 30 >
     V AND P < 65) OR P = 32 THEN
     2000
1100  IF P < 65 THEN  VTAB P - 30
     : HTAB 1:LE = 19: GOSUB 500:
     LE = 5
1120  IF P > 64 THEN  VTAB P - 62
     : HTAB 21:LE = 39: GOSUB 500
     :LE = 25
1140  INVERSE : HTAB LE:FT$ =  LEFT$
     (T$,1):T$ =  RIGHT$ (T$,16):
      PRINT T$: NORMAL
1160  IF  ASC (FT$) = 193 THEN C$
      = "RUN"
1180  IF  ASC (FT$) = 212 THEN C$
      = "EXEC"
1200  IF  ASC (FT$) = 201 THEN C$
      = "RUN"
1220  IF  ASC (FT$) = 194 THEN C$
      = "BRUN"
1240  PRINT : PRINT  CHR$ (4);C$;
     T$: END
2000  POKE 216,0: HOME : INPUT "Y
     OU HAVE MADE AN INCORRECT EN
     TRY-PRESS RETURN TO CONTINUE
     :";T$: RUN
3000  DATA   1428,15,1439,234,144
     0,234,1441,234,1378,234,1379
     ,234,1380,234,1403,234,1404,
     234,1405,234,1416,32,1417,66
     ,1418,174,1421,234,1422,234,
     1423,234,1320,234,1321,234,1
     322,234,1347,234,1348,234,13
     49,234,1313,51,1503,48
```

香港微型電腦訓練中心

招　生

教署立案
(1980年)

① 本校師資優良，課程由教育司署認可之電腦專業人仕任教。

② 本校置有十多部高速優良美國電腦系統。

③ 除正式授課及實習外，另提供自由時間免費實習電腦，加強訓練。

④ 本校每月均有新班，每星期上課一或兩次，時間分上、下午或晚班制。

暑期電腦課程：

(1) (初級)Computer Concepts+BASIC Programming + Operation($315) 25小時

(2) Advanced-BASIC Programming($340) 25小時

(3) COBOL($360)30小時

(4) FORTRAN($360) 30小時

(5) PASCAL($360) 30小時

(6) Assembly Language($360) 30小時

(7) CP/M & MP/M(MULTI-USERS) OPERATING SYSTEMS($295) 20小時

(8) Word-processing, Supercalc, Multiplan and dBASE II ($450) 15小時

校址：九龍佐敦道渡船街1-4號金霞閣A座②字樓

（碼頭行人天橋側，大家樂鄰）

詢問電話：3-858615　　3-329996

任何保護方法皆行
最簡單的拆帶程式

讀者如有興趣試將此程式改良，令到在轉錄資料時，螢幕上可清楚指示當時處於何種狀態。

對於優異的改良程式，我們將送他一隻程式磁碟。投稿辦法可參看〔投稿需知〕

對於一些錄音機的用家來說，他們都有很多拆帶的方法，而且也有不少書籍都曾經對錄音帶的保護提出破解的方法。但是，那些保護不但種類繁多，而且又要買 RAM CARD 或改變監督程式才能進行，對一些初學者來說，實在是一件十分頭痛的事。所以，筆者特別介紹一個拆帶程式，程式能將在磁帶上的資料完全搬去另一盒磁帶上，而失眞率幾乎零，所以以後各位錄音機的用家便不需要學拆帶了。

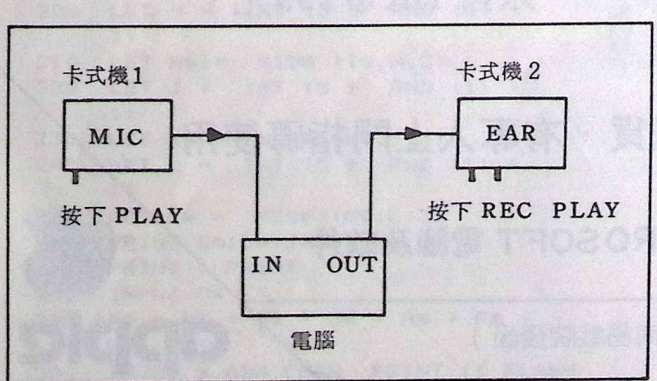
本文的程式十分簡單，大致上是從 INPUT埠拿取資料，再從 OUTPUT 埠輸出輸料。可是，原本在磁帶的資料怎樣辦，再說錄音機也不能同時讀出資料和寫上資料的，所以，我們便要用兩部錄音機了。讀者們不要以爲這和普通的錄音帶一樣，因爲它是由電腦讀出磁帶上的資料，再由電腦寫上磁帶，除非是硬件部份出現問題，否則是一定成功的。

程式是有兩個版本的，MACHIN—E CODE 品 APPLESOFT。MACHINE CODE 程式中 C060 是由磁帶讀出資料，而 C020 則是輸出資料，寫上磁帶。在程式運行中，有資料輸入的話，則立即輸出到另一部錄音機，沒有資料輸入的話則再跳因前面，等代下一個資料輸入，這樣不斷的進行着，就是這程式的工作過程。

當程式執行時，螢光幕上會一點動靜都沒有，那麼，怎樣才可知道是否錄完呢？我們可以在 INPUT 那條綫上接上一個 TESTER 就可以了。

```
15  REM  ***************
25  REM  * TAPE  COPY *
35  REM  *      BY     *
45  REM  * PATRICK HO *
55  REM  *      &      *
65  REM  *  CHECKED    *
75  REM  *      BY     *
85  REM  *LEE WEI MAN.*
95  REM  ***************
105  HOME
115  DATA 173,96,192,16,251,173,3
     2,192,173,48,192,173,96,192,
     48,251,173,32,192,173,48,192
     ,76,0,3,0
125  FOR A = 768 TO 793: READ B: POKE
     A,B: NEXT
135  PRINT "TYPE 300G TO START"
145  CALL  - 151
```

```
]BLOAD BINARY TAPE COPY
]CALL-151

*300L

0300-   AD 60 C0    LDA   $C060
0303-   10 FB       BPL   $0300
0305-   AD 30 C0    LDA   $C030
0308-   AD 60 C0    LDA   $C060
030B-   30 FB       BMI   $0308
030D-   AD 20 C0    LDA   $C020
0310-   4C 00 03    JMP   $0300
0313-   00          BRK
0314-   00          BRK
0315-   00          BRK
0316-   00          BRK
0317-   00          BRK
0318-   00          BRK
0319-   00          BRK
031A-   00          BRK
031B-   00          BRK
031C-   00          BRK
031D-   00          BRK
031E-   00          BRK
031F-   00          BRK
*
```



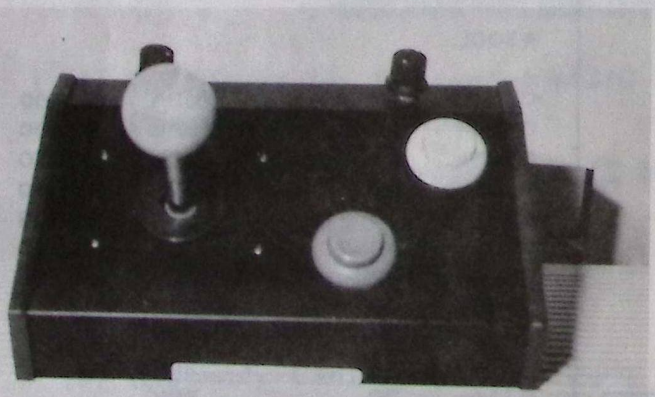
1·2·3 COMPUTER CENTRE 電腦中心

〈專營〉

多款 電腦硬件，軟件，配件，零件，盒帶，磁碟，書籍，保養及維修。

—特設—

訂造 JOYSTICK，款式大小任君選擇，適合任何電腦，比普通耐用十倍，緊張刺激，仿似身在遊戲中心一樣。保用一年 。



• 收購 •

二手電腦，任何牌子，磁碟機，顯示器或其他附件一律歡迎，全港最高，不妨試試。

（歡迎查詢）

隆重介紹 黑戰士 型 JOYSTICK，介紹期內只售 HK $199元。

凡購買電腦一套，隨即免費送貨，有專人上門指導使用

代客 訂購名廠原裝 IBM-APPLE II-MICROSOFT 電腦及軟件

地址：深水埗南昌街白田購物中心一樓 101 室（南昌戲院後面）
電話：3-7783923

apple

Apple is the trademark of Apple Computer Inc.

TYPING TRAINING

Y.M.S.

有打字機或電腦的朋友都知道，初學打字是非常困難的，學的時候害怕手勢和姿勢不對。一邊學一邊看打字書練習，甚爲不便。本文所介紹的程式，可助他們一臂之力及解決在練習打字上所遇到的重重困難。

程式非常簡單，當程式運行時，首先螢幕會出現二個要點，接着選擇難度——（1）最易；（2）比較難；（3）最難，包括數目字。電腦將指定難度的文字 RND 後，印在螢幕上，只要我們照着螢幕上的文字輸入，打錯的話，電腦將你打錯的文字與正確比較。打對的話，螢幕印出二個問題，讓我們來答是或否。其實，簡單來說，程式的目的，是在改良我們的打字速度和減少打錯字。□

```
10  REM  ******************
20  REM  *TYPING TRAINING*
30  REM  *    Y.M.S.      *
40  REM  * JUNE,1984.     *
50  REM  *                *
60  REM  ******************
70  REM  **** THIS IS A TRAIN TYP
     EWRITE GAME ****
80  HOME : VTAB 2: HTAB 4: FLASH
     : PRINT "*THIS A TRAIN TYPEW
     RITE GAME *": NORMAL
85  PRINT : FOR I = 1 TO 40: PRINT
     "-";: NEXT I
90  VTAB 6: PRINT "Type the latte
     rs in one line with"
100  VTAB 8: PRINT "No space betw
     een letters"
105  PRINT : FOR I = 1 TO 40: PRINT
     "-";: NEXT I
110  VTAB 13: HTAB 3: PRINT "What
      level of difficulty?(1-3)";
     : GET A: PRINT A
120  ON A GOSUB 430,460,490
130  PRINT
140  LET B =  INT (5 *  RND (1) +
     1)
150  LET B$ =  MID$ (C$,B,3)
160  LET D =  INT (5 *  RND (1) +
     1)
170  LET D$ =  MID$ (E$,D,3)
180  LET F =  INT ( RND (1) * 5) +
     1
190  LET F$ =  MID$ (G$,F,3)
200  LET H =  INT (5 *  RND (1) +
     1)
210  LET H$ =  MID$ (I$,H,3)
220  LET J =  INT (5 *  RND (1) +
     1)
230  LET J$ =  MID$ (K$,J,3)
240  LET L =  INT (5 *  RND (1) +
     1)
250  LET L$ =  MID$ (M$,L,3)
260  PRINT B$;D$;J$;H$;F$;L$
270  PRINT : PRINT
280  INPUT O$
290 AB$ = B$ + D$ + J$ + H$ + F$ +
     L$
300  IF O$ = AB$ THEN  PRINT :: FLASH
     : HTAB 13: PRINT "<<VERY GOO
     D>>": NORMAL : GOTO 370
310  IF O$ <  > AB$ THEN  GOTO 32
     0
320  PRINT : INVERSE : HTAB 5: PRINT
     "<<TYPE ERROR>>": NORMAL : PRINT

330  FOR B = 1 TO 18 STEP 3
340  IF  MID$ (O$,B,3) <  >  MID$
     (AB$,B,3) THEN  PRINT  MID$
     (O$,B,3);: PRINT "";: FLASH
     : PRINT "MUST ";: NORMAL : PRINT
     "";: FOR Z = 1 TO 140 / 2:S =
      PEEK ( - 16336): NEXT Z: PRINT
      MID$ (AB$,B,3): PRINT : PRINT

350  NEXT B
360  HTAB 5: INVERSE : PRINT "<<T
     YPE AGAIN>>": FOR PAUSE = 1 TO
     1000: NEXT PAUSE: NORMAL : GOTO
     130
370  PRINT : PRINT "Would you lik
                e same level aga
     in?(Y/N)";: GET Q$: PRINT : IF
     Q$ = "Y" THEN  GOTO 130
380  IF Q$ = "N" THEN  GOTO 390
390  PRINT : PRINT "Would you lik
     e different level?(Y/N)";: GET
     R$: PRINT
400  IF R$ = "Y" THEN  PRINT : GOTO
     70
410  IF R$ = "N" THEN  GOTO 420
420  CALL  - 1184: END
430  PRINT
440 C$ = "FALG":E$ = "KSDJ":G$ =
     "IESTR":K$ = "RHTA":I$ = "EL
     FI":M$ = "JUEY"
450  RETURN
460  PRINT
470 C$ = "PQWL":E$ = "OVFN":G$ =
     "MCKB":K$ = "CQPZ":I$ = "MXF
     T              ":M$ = "KOED"
480  RETURN
490  PRINT
500 C$ = "1FDOO":E$ = "F382J9":G$
      = "4HLOH5":K$ = "ZQP6LKF":I
     $ = "YMS7CMT":M$ = "J6HTSQZ7
     "
510  RETURN                    □
```

香港教育司署立案（1957年創校）

世界函授學校

規模大　優點多　科目好

初級電腦科
電腦基本原理，電腦操作，邏輯電路，程式編寫八個月畢業。

初級無綫電科
全部圖解教導，原理裝修並重，極易學習，十個月畢業。

半導體裝修科
配合進步的技術，具初學或職業的裝修人員必修的學科十個月畢業。

實用數理
可以加強數理基礎，費用少，收效大，六個月畢業。

實用商科
著重實際應用，加強商業學識，期間短，學習易，六個月畢業。

無綫電工程
設計，研究和深造的理想學科。一年半畢業。

汽車工程
適合職業及業餘人員加強對汽車方面之學識。

冷氣工程
冷氣，冷藏人員缺乏，專門學校極少，一年半畢業。

商業美術科
內容新，範圍廣，包括美術和最新廣告設計。一年畢業。

應用漫畫
漫畫可訓練頭腦，增強思想，取材新，學習易，六個月畢業。

導師
國內外著名大學學士，經香港教育司考核合格，教導認眞，決不馬虎。

講義
全部經教育司署審定，閱讀容易，印刷精美，免費供應。

費用
除低廉之學費外，學生決無額外負擔。

文憑
本校頒發文憑，均經政府簽証，實是最佳學歷証件。

校址： 香港九龍彌敦道五六六號9樓A座
電話：③—803532

本港學生可利用電話詢問索取章程　電話③－803532

請依綫剪下，填好姓名，回郵地址，貼在信封上投寄，本校即免費寄上簡章

郵件請寄：P.O.BOX 70011, KOWLOON, CENTRAL POST OFFICE
九龍中央郵政局信箱 70011 號

技術就是財產　　學問就是力量

（寄自）

姓名：

地址：

（請註明要學的科目）

- □初級電腦科
- □無綫電工程科
- □初級無綫電科
- □半導體裝修科
- □實用數理科
- □汽車工程
- □冷氣工程
- □實用商科
- □商業美術
- □應用漫畫

請貼上郵票

香港教育司署立案

To：世界函授學校註册處

P. O. BOX No. 70011

KOWLOON CENTRAL POST OFFICE,

HONG KONG.

香港九龍中央郵局信箱70011號

電話：K-803532

APPLESOFT 滙編器

□□

由於很多讀者都不明白在雜誌上刊登的滙編語言程式如何鍵入，因此今期爲各位介紹一個由BASIC寫成的滙編器，它是由兩個獨立的程式組成。事實上它並非是很理想的程式，不過由於它的使用簡單而且程式長度短，因此可以節省寶貴的記憶空間，以供滙編程式的貯存，是有一定價值的。

滙編程式的敍述句格式

程式中敍述句是採用標準的6502記憶語詞，其格式必須要有如下的格式：

1、第一至第8格是存放標記（LABEL）的（如有的話）。

2、第10至第12格是存放指令用的（即運算元）。

3、任何運算碼必須由第14格開始。

4、註譯可放在30格開始。

滙編器的基本運算元

除基本的運算元之外，此滙編器還增加多了三個：

1、ORG ——它用來表示程式的起始位置。

2、EQU ——它用來表示標記（LABEL）相當於指定的記憶體位置。

3、DRS ——它是 DEEINE RELOCATABLE STORAGE 的縮寫，作用相似於 EQU，但它能保留特定的空間來作爲數據的貯存之用，例如 TOTAL DRS 2 ，使能在程式開始時保留2個 BYTE 給標記 TOTAL 用。

另一個使用法是將一個或以上的初始值存入，例如在保留的 BYTE 數目後緊隨一個感嘆號（!）再跟著一些你希望存在記憶體內該標記的字串，而結尾同樣加上一個感嘆號（!）作爲結束。例如：

```
PRNTLINE DRS 6!ABCDEF!
```

滙編器會將這些字串翻譯成它們相對的 ASCII 數碼，並且一個接一個地放在要註明的ORG 敍述句的地址上，這樣在編寫一些印表的程式便簡易多了。以上的 DRS 及 EQU必須緊隨於 ORG 後。

滙編器的特點

很多情況下有需要知道一個標記的眞正位址，例如把一個位址的數據移往一個子程式上時爲然，爲此本滙編器加入兩種不同的變化再作配合此一用途，所採用的符號是"小於"符號（<）來註明該位址的高位 BYTE 及（>）"大於"符號來註明該位址的低位 BYTE 。例如下列的程式：

```
TABLE DRE 10!ABCDEFGHIJ!
      LDA <TABLE
      LDY >TABLE
```

此例能將 TABLE 的高位 BYTE 載入累加器中，而低位 BYTE 則載入往Y暫存器裡。注意，這裡並沒有把TABLE內含的值載入往以上兩個地方，只是把其位址載入吧了，它所產生的 OP 代碼與立即地址模式所產生的一樣。

本滙編程式共有兩個，分別爲 APPLESOFT ASSEMBLER，及 ASSEMBLER EDITOR。

ASSEMBLER EDITOR 能把閣下鍵入的程式轉爲一個 TEXT FILE 存入磁碟裏，而此檔案在較後是給予滙編器翻譯成機械碼用的。

利用這個編輯器可以作滙編程式的改良或修改之用。它首先會詢問你這是否一個新檔案，若它早已存在的話，便從磁碟載入並進入編輯狀態下，若它是新檔案的話，便進入輸入狀態下。

①在輸入或編輯狀態下，ESC 鍵的作用是令游標的位置跳往第10格上，以幫助運算元的起始位置。

②若不輸入任何東西而按下 RETURN鍵會令輸入狀態變爲編輯狀態，或編輯狀態變爲輸入狀態。

③在編輯狀態下，下列的各指令可以產生不同功能的效果：

n	—編輯第n行
L	—列出整個滙編程式
Ln	—列出第n行
Ln,m	—列出第n行至第m行
Dn	—刪除第n行
In	—進入輸入狀態，並加插一些東西往n行之後
P	—印出整個程式往列表機上。
Pn	—印出第n行往列表機上
Pn,m	—印出第n行至號m行往列表機上
?	—顯示出各指令的用途。

④在任何時候，ESC 及→鍵可以令你到達任何位置，但緊記若程式的該行須用→鍵來重抄入記憶裏才可按 RETURN鍵。

⑤CTRL—X 可令所鍵入的東西取消並重新再輸入。

⑥在編輯狀態裡鍵入 SAVE 會把程式貯入磁碟裏。

使用

至於第二個滙編程式 APPLESOFT ASSEMBLER 經由編輯器以 SAVE 命令輸入後，便會問閣下是否需要該程式組合成爲機械碼，若回答是的話，它便將該檔案讀入並作組合過程。

滙編程式會執行兩次核驗過程，之後會顯示出每個指令的號碼，若期間出現有錯誤，它會給出一個錯誤訊息並予你重新運作編輯程式以修正錯誤。假如一切順利，所產生的機械程式的起始位址及長度會顯示出來。順帶一提的是有些滙編程式的位址會與DOS 所佔用的位址相衝突，此時滙編程式會將它所組合後的機械碼位址移往不相衝突的地方，並將

它的位址顯示出來，留待閣下在使用時重新移往所需的位置上去。

例子

現試舉一個使用例子，〔NOTE〕是一個能揚聲器發出聲響的子程式。

1、LOAD ASSEMBLER EDITOR 並鍵入 RUN。
2、現在編輯程式會問你滙編程式的名字，鍵入"NOTE"。
3、這是一個新的程式，因此在以下的問題裏回答"Y"。
4、現在會進入輸入狀態，把列表（1）的NOTE 程式鍵入，儘量利用ESC 鍵來定位。
5 當完全鍵入 NOTE 程式後，停止再按入任何東西，按RETURN鍵，這樣便進入編輯狀態。
6、按L檢查所鍵入的有沒有錯誤，若有按入要更改的行號。
7、若沒有錯誤，按 SAVE 把程式貯入磁碟裏。
8、在電腦顯示"DO YOU WISH TO ASSEMBL－E YOUR FILE?"這問題時回答Y。
9、這時電腦會問你所要滙編的程式名字，鍵入 NOTE 。
10、現在滙編器會進行組合工作，它會把指令顯示出來。
11、程式正確無誤，滙編器會把程式的開始位址及長度顯示出來，並把機械碼移往十進制 20000 的位址並以BSAVE NOTE.OBJ 存入磁碟內。（即 $4E20）
12、在此時可以按 CALL－151 進入監督程式按 4E20 L（參看表二），便可到該程式的機械碼。
13、若要執行此程式，可以按 RESET 進入 BASIC 並鍵入 CALL 20000 ，或在監督程式裏，按 4E 20 G ，在兩種情形下，你都可以聽到揚聲器發出一聲音。祝各位一切順利，再會！□

列表（1）NOTE 子程式

```
          位置1          位置10  位置13
          1:             ORG  20000
          2:SPEAKER      EQU  $C030
          3:NOTE         EQU  768
          4:DUR          EQU  769
          5:             LDA  #200
          6:             STA  NOTE
          7:             STA  DUR
          8:L1           LDA  SPEAKER
          9:L2           DEY
          10:            BNE  L3
          11:            DEC  DUR
          12:            BEQ  L4
          13:L3          DEX
          14:            BNE  L2
          15:            LDX  NOTE
          16:            JMP  L1
          17:L4          RTS
```

表（2） 經滙編後的程式列表

```
*4E20L

4E20-   A9 C8        LDA   #$C8
4E22-   8D 00 03     STA   $0300
4E25-   8D 01 03     STA   $0301
4E28-   AD 30 C0     LDA   $C030
4E2B-   88           DEY
4E2C-   D0 05        BNE   $4E33
4E2E-   CE 01 03     DEC   $0301
4E31-   F0 09        BEQ   $4E3C
4E33-   CA           DEX
4E34-   D0 F5        BNE   $4E2B
4E36-   AE 00 03     LDX   $0300
4E39-   4C 28 4E     JMP   $4E28
4E3C-   60           RTS
4E3D-   FF           ???
```

```
10  REM  **********************
15  REM  *                    *
20  REM  *                    *
25  REM  * ASSEMBLER.EDITOR   *
30  REM  *                    *
35  REM  * COPYRIGHT(C) 1984  *
40  REM  *                    *
45  REM  *                    *
50  REM  **********************
55  REM
150 D$ =  CHR$ (4)
160  DIM A$(250),A(250)
170  HOME : VTAB 22: PRINT "** CO
     PYRIGHT 1984 BY COMPUHOUSE,I
     NC.**"
180  VTAB 2: INVERSE : PRINT "ASS
     EMBLER EDITOR": NORMAL
190  GOSUB 1440: ONERR  GOTO 1600
200  VTAB 6: CALL  - 958: INPUT "
     ENTER FILE NAME:";PR$: IF  LEN
     (PR$) = 0 OR  LEN (PR$) > 15
      OR  VAL (PR$) > 0 THEN 200
210  VTAB 8: CALL  - 958: PRINT D
     $"OPEN "PR$: PRINT D$"CLOSE"

220  VTAB 8: INPUT "IS THIS A NEW
      FILE?(Y/N):";NP$
230  IF NP$ = "N" THEN  GOSUB 109
     0: GOTO 260
240  IF NP$ = "Y" OR NP$ = "" THEN
      PRINT : GOTO 280
250  VTAB 8: CALL  - 958: GOTO 22
     0
260  PRINT : PRINT "EDIT"
270  GOSUB 850
280  IF L$ = "" THEN L$ = "L ": GOSUB
```

APPLESOFT 滙編器

```
     370: GOTO 260
290  IF  VAL (L$) <  > 0 THEN  GOSUB
     490: GOTO 260
300  IF  LEFT$ (L$,2) = "D " THEN
      GOSUB 570: GOTO 260
310  IF  LEFT$ (L$,1) = "L" THEN
      GOSUB 690: GOTO 260
320  IF  LEFT$ (L$,1) = "P" THEN
      PRINT D$"PR#1": GOSUB 690: PRINT
     D$"PR#0":L$ = "L ": GOTO 280

330  IF  LEFT$ (L$,1) = "?" THEN
      GOSUB 1350: GOTO 260
340  IF  LEFT$ (L$,2) = "I " THEN
      GOSUB 370: GOTO 260
345  IF  LEFT$ (L$,3) = "ASM" THEN
      PRINT D$"RUN APPLESOFT.ASSE
     MBLER"
350  IF  LEFT$ (L$,4) = "SAVE" THEN
     1200
360  PRINT "INVALID COMMAND ": GOTO
     260
370  REM
380  PRINT "INPUT"
390 N = 0:NL = 1: IF  LEFT$ (L$,2
     ) = "I " THEN NL =  VAL ( MID$
     (L$,3)):N = NL - 1
400  IF A(N) <  > 0 AND NL <  VAL
     ( MID$ (L$,3)) + 1 THEN N =
     A(N):NL = NL + 1: GOTO 400
410  PRINT NL":"
420  GOSUB 850
430  IF L$ = "" THEN  RETURN
440 J = NL
450  IF A$(J) <  > "" THEN J = J +
     1: GOTO 450
460 A$(J) = L$
470 A(J) = A(N):A(N) = J:N = J:NL
     = NL + 1
480  GOTO 410
490  REM
500 N = 0
510  FOR J = 1 TO  VAL (L$)
520  IF A(N) = 0 THEN  PRINT "LIN
     E NOT FOUND": RETURN
530 N = A(N)
540  NEXT
550  PRINT N;":": PRINT A$(N);:L$
      = A$(N) +  LEFT$ ("

         ",40 -  LEN (A$(N))):: HTAB
     1: GOSUB 900: IF L$ = "" THEN
      RETURN
560 A$(N) = L$: RETURN
570  REM
580 LN =  VAL ( MID$ (L$,2)) - 1
590  IF LN < 0 THEN  PRINT "LINE
     NOT  FOUND ": RETURN
600 N = 0
610  IF LN = 0 THEN 660
620  FOR J = 1 TO LN
630  IF A(N) = 0 THEN  PRINT "LIN
     E NOT FOUND": RETURN
640 N = A(N)
650  NEXT
660 A$(A(N)) = ""
670 A(N) = A(A(N))
680  RETURN
690  REM
700  IF  LEN (L$) < 3 THEN LS = 0
     :LE = 0: GOTO 770
710 LS =  VAL ( MID$ (L$,3))
720  IF  LEN (L$) < 4 THEN 760
730  FOR K = 4 TO  LEN (L$)
740  IF  MID$ (L$,K,1) = "," OR  MID$
     (L$,K,1) = " " THEN LE =  VAL
     ( MID$ (L$,K + 1)): GOTO 770

750  NEXT
760 LE = LS
770  IF LE = 0 THEN LE = 99999
780 N = 0
790  FOR J = 1 TO LE
800 N = A(N)
810  IF N = 0 THEN 840
820  IF J >  = LS THEN  PRINT J;"
     :";A$(N)
830  NEXT
840  RETURN
850  REM
860 L$ = "
                              "

900 CC = 1
910  IF CC < 1 THEN CC = 1
920  IF CC > 39 THEN CC = 39
930  HTAB CC: GET KB$
940  IF  ASC (KB$) = 8 THEN CC =
     CC - 1: GOTO 910
950  IF  ASC (KB$) = 21 THEN CC =
     CC + 1: GOTO 920
960  IF  ASC (KB$) = 13 THEN 1040

970  IF  ASC (KB$) = 24 THEN  PRINT
      CHR$ (92): GOTO 860
980  IF  ASC (KB$) = 27 THEN CC =
     10: GOTO 930
990  ON  ASC (KB$) < 32 GOTO 930:
      PRINT KB$;
1000  IF CC = 1 THEN L$ = KB$ +  MID$
     (L$,2):CC = CC + 1: GOTO 930

1010  IF CC = 39 THEN L$ =  LEFT$
```

```
     (L$,38): HTAB CC: PRINT " ";
     : GOTO 930
1020 L$ =  LEFT$ (L$,CC - 1) + KB
     $ +  MID$ (L$,CC + 1)
1030 CC = CC + 1: GOTO 930
1040  IF CC = 1 THEN L$ = "": GOTO
     1060
1050 L$ =  LEFT$ (L$,CC - 1)
1060  HTAB 1: PRINT L$; SPC( 40 -
     C)
1070  PRINT
1080  RETURN
1090  REM
1100 A(0) = 1: ONERR  GOTO 1600
1110  PRINT D$"OPEN "PR$
1120  PRINT D$"READ "PR$
1130 N = N + 1
1140  INPUT A$(N)
1150  IF  MID$ (A$(N),10,3) = "EO
     F" THEN A$(N) = "":N = N - 1
     :A(N) = 0: GOTO 1180
1160 A(N) = N + 1
1170  GOTO 1130
1180  PRINT D$"CLOSE"PR$: POKE 21
     6,0
1190  RETURN
1200  REM
1210  ONERR  GOTO 1600
1220  PRINT : PRINT D$"OPEN "PR$:
      PRINT D$"CLOSE": PRINT D$"D
     ELETE"PR$
1230  PRINT D$"OPEN "PR$
1240  PRINT D$"WRITE "PR$
1250 N = 0
1260  PRINT  CHR$ (34)A$(A(N)) CHR$
     (34)
1270 LN = N
1280 N = A(N)
1290  IF A(N) <  > 0 THEN 1260
1300  IF  MID$ (A$(LN),10,3) <  >
     "EOF" THEN  PRINT  CHR$ (34)
     "          EOF" CHR$ (34)
1310  PRINT D$"CLOSE"PR$
1320  INPUT "DO YOU WISH TO ASSEM
     BLE YOUR FILE?(Y/N)";K$: IF
     K$ = "Y" THEN  PRINT D$"RUN
     APPLESOFT.ASSEMBLER"
1330  HOME : POKE 216,0: END
1350  HOME : INVERSE : PRINT "COM
     MANDS:                ": NORMAL

1360  PRINT "L        LIST ENTIRE
     PROGRAM"
1370  PRINT : PRINT "L N    LIST
     LINE N"
1380  PRINT : PRINT "L N,M  LIST
     N THROUGHM"
1390  PRINT : PRINT "D N  DELETE
     LINE N"
1400  PRINT : PRINT "I N  INPUT L
     INES AFTER N"
1410  PRINT : PRINT "P(N(,M)) PRI
     NT LISTING"
1420  PRINT : PRINT "SAVE   SAVE
     THE PROGRAM TO DISK"
1430  PRINT : PRINT "N   EDIT LIN
     E N"
1434  PRINT : PRINT "<ESC>  TAB (
     ONLY IN EDIT OR INPUT)"
1436  PRINT : PRINT "?    THIS HE
     LP SCREEN": RETURN
1440  DIM PN(16)
1450 PN(1) = 67:PN(2) = 79:PN(3) =
     77:PN(4) = 80
1460 PN(5) = 85:PN(6) = 84:PN(7) =
     73:PN(8) = 78
1470 PN(9) = 71:PN(10) = 32:PN(11
     ) = 45:PN(12) = 32
1480 PN(13) = 65:PN(14) = 71:PN(1
     5) = 69:PN(16) = 32
1500  FOR VN = 1 TO 16
1510  VTAB 4
1520  FOR VO = VN TO 1 STEP  - 1
1530  PRINT  CHR$ (PN(17 - VO));
1540  NEXT VO
1550  PRINT
1560  NEXT VN
1570  PRINT
1580  RETURN
1600  POKE 216,0: PRINT : PRINT D
     $"CLOSE"
1610  IF  PEEK (222) = 5 THEN  PRINT
     D$"DELETE"PR$: PRINT  CHR$ (
     7)"FILE NOT ON THIS DISK ": PRINT
     "PRESS ANY KEY=>";: GET A$: PRINT
     : RUN
1620  IF  PEEK (222) = 16 THEN  PRINT
      CHR$ (7)"INVALID FILE NAME.
     PRESS ANY KEY=>";: GET A$: PRINT
     : RUN
1630  IF  PEEK (222) = 6 THEN  PRINT
     "PUT IN DISK WITH APPLESOLF
     ASSEMBLER": PRINT "AND PRESS
      ANY KEY+>";: GET A$: PRINT
     : GOTO 1310
1640  PRINT  CHR$ (7)"DISK ERROR
     #"; PEEK (222);"IN LINE"; PEEK
     (218) +  PEEK (219) * 256
1650  END
```

APPLESOFT 滙編器

```
1  REM  ************************
2  REM  *                      *
3  REM  *                      *
4  REM  * APPLESOFT.ASSEMBLER *
5  REM  *                      *
6  REM  *                      *
7  REM  * COPYRIGHT (C) 1984  *
8  REM  *                      *
9  REM  ************************
10  DATA ADC,0,109,101,105,125,12
    1,97,113,117,0,0,0
20  DATA ,0,45,37,41,61,57,33,49,
    53,0,0,0
30  DATA  ASL,10,14,6,0,30,0,0,0,
    22,0,0,0
40  DATA  BCC,0,0,0,0,0,0,0,0,0,0
    ,144,0
50  DATA   BCS,0,0,0,0,0,0,0,0,0,
    0,176,0
60  DATA   BEQ,0,0,0,0,0,0,0,0,0,
    0,240,0
70  DATA   BIT,0,44,36,0,0,0,0,0,
    0,0,0,0
80  DATA   BMI,0,0,0,0,0,0,0,0,0,
    0,48,0
90  DATA   BNE,0,0,0,0,0,0,0,0,0,
    0,208,0
100  DATA   BPL,0,0,0,0,0,0,0,0,0
    ,0,16,0
110  DATA   BRK,0,0,0,0,0,0,0,0,0
    ,0,0,0
120  DATA   BVC,0,0,0,0,0,0,0,0,0
    ,0,80,0
130  DATA   BVS,0,0,0,0,0,0,0,0,0
    ,0,112,0
140  DATA   CLC,24,0,0,0,0,0,0,0,
    0,0,0,0
150  DATA   CLD,216,0,0,0,0,0,0,0
    ,0,0,0,0
160  DATA   CLI,88,0,0,0,0,0,0,0,
    0,0,0,0
170  DATA   CLV,184,0,0,0,0,0,0,0
    ,0,0,0,0
180  DATA   CMP,0,205,197,201,221
    ,217,193,209,213,0,0,0
190  DATA   CPX,0,236,228,224,0,0
    ,0,0,0,0,0,0
200  DATA   CPY,0,204,196,192,0,0
    ,0,0,0,0,0,0
210  DATA   DEC,0,206,198,0,222,0
    ,0,0,214,0,0,0
220  DATA   DEX,202,0,0,0,0,0,0,0
    ,0,0,0,0
230  DATA   DEY,136,0,0,0,0,0,0,0
    ,0,0,0,0
240  DATA   EOR,0,77,69,73,93,89,
    65,81,85,0,0,0
250  DATA   INC,0,238,230,0,254,0
    ,0,0,246,0,0,0
260  DATA   INX,232,0,0,0,0,0,0,0
    ,0,0,0,0
270  DATA   INY,200,0,0,0,0,0,0,0
    ,0,0,0,0
280  DATA   JMP,0,76,0,0,0,0,0,0,
    0,0,0,0,108
290  DATA   JSR,0,32,0,0,0,0,0,0,
    0,0,0,0
300  DATA   LDA,0,173,165,169,189
    ,185,161,177,181,0,0,0
310  DATA   LDX,0,174,166,162,0,1
    90,0,0,0,182,0,0
320  DATA   LDY,0,172,164,160,188
    ,0,0,0,180,0,0,0
330  DATA   LSR,74,78,70,0,94,0,0
    ,0,86,0,0,0
340  DATA   NOP,234,0,0,0,0,0,0,0
    ,0,0,0,0
350  DATA   ORA,0,13,5,9,29,25,1,
    17,21,0,0,0
360  DATA   PHA,72,0,0,0,0,0,0,0,
    0,0,0,0
370  DATA   PHP,8,0,0,0,0,0,0,0,0
    ,0,0,0
380  DATA   PLA,104,0,0,0,0,0,0,0
    ,0,0,0,0
390  DATA   PLP,040,0,0,0,0,0,0,0
    ,0,0,0,0
400  DATA   ROL,42,46,38,0,62,0,0
    ,0,54,0,0,0
410  DATA   ROR,106,110,102,0,126
    ,0,0,0,118,0,0,0
420  DATA   RTI,64,0,0,0,0,0,0,0,
    0,0,0,0
430  DATA   RTS,96,0,0,0,0,0,0,0,
    0,0,0,0
440  DATA   SBC,0,237,229,233,253
    ,249,225,241,245,0,0,0
450  DATA   SEC,56,0,0,0,0,0,0,0,
    0,0,0,0
460  DATA   SED,248,0,0,0,0,0,0,0
    ,0,0,0,0
470  DATA   SEI,120,0,0,0,0,0,0,0
    ,0,0,0,0
480  DATA   STA,0,141,133,0,157,1
    53,129,145,149,0,0,0
490  DATA   STX,0,142,134,0,0,0,0
    ,0,0,150,0,0
500  DATA STY,0,140,132,0,0,0,0,0
    ,148,0,0,0
510  DATA   TAX,170,0,0,0,0,0,0,0
    ,0,0,0,0
520  DATA   TAY,168,0,0,0,0,0,0,0
    ,0,0,0,0
530  DATA   TSX,186,0,0,0,0,0,0,0
    ,0,0,0,0
540  DATA   TXA,138,0,0,0,0,0,0,0
    ,0,0,0,0
550  DATA   TXS,154,0,0,0,0,0,0,0
    ,0,0,0,0
560  DATA   TYA,152,0,0,0,0,0,0,0
    ,0,0,0,0
570  HOME : VTAB 22: PRINT "** CO
    PYRIGHT 1984 BY COMPUHOUSE,I
    NC.**"
580  VTAB 3: INVERSE : PRINT "APP
    LESOLF ASSEMBLER": NORMAL : PRINT

590  GOSUB 2270
```

APPLESOFT 滙編器

```
600  CALL  - 958
610 SA = 30208
620  HIMEM: SA
630 D$ =  CHR$ (4): VTAB 7: CALL
     - 958: INPUT "SOURCE FILE T
    O LOAD:";PR$: IF  LEN (PR$) =
    0 OR  LEN (PR$) > 15 OR  VAL
    (PR$) > 0 THEN 630
640  ONERR  GOTO 660
650  PRINT D$"VERIFY";PR$: PRINT
    D$"OPEN";PR$: PRINT D$"READ"
    ;PR$: POKE 216,0: GOTO 670
660  POKE 216,0: PRINT  CHR$ (7)"
    THAT FILE IS NOT ON THIS DIS
    K .": PRINT "PRESS ANY KEY T
    O CONTINUE. ": GET K$: PRINT
    : GOTO 630
670  INPUT IN$: IF  MID$ (IN$,10,
    3) = "EOF" THEN  PRINT D$"CL
    OSE": GOTO 710
680 NI = NI + 1
690  IF  LEFT$ (IN$,8) <  > "
         " THEN NL = NL + 1
700  GOTO 670
710  DIM OP(56,12),OP$(56)
720  DIM IN$(NI),MD(NI),LB$(NI),T
    P$(NI),LN(NI),LO(NI + 1)
730  DIM LF$(NL),AF(NL),LR$(NL),D
    R(NL)
740  RESTORE
750  VTAB 12: CALL  - 958: PRINT
    "FIRST PASS:"
760  PRINT D$"OPEN "PR$: PRINT D$
    "READ"PR$
770  INPUT LA$: IF  MID$ (LA$,10,
    3) <  > "ORG" OR  LEN (LA$) <
    14 THEN  PRINT : PRINT D$"CL
    OSE "PR$: PRINT  CHR$ (7): PRINT
    "SOURCE CODE MUST BEGIN WITH
    ' ORG'": GOTO 2190
780 N = 1:HX$ =  MID$ (LA$,14): IF
     LEFT$ (HX$,1) = "$" THEN  GOSUB
    2490:LA = DT: GOTO 800
790 LA =  VAL (HX$)
800  FOR N = 2 TO NI
810  INPUT IN$
820  IF  LEN (IN$) > 14 THEN HX$ =
     MID$ (IN$,14): GOSUB 2490:I
    N$ =  LEFT$ (IN$,13) + HX$
830  VTAB 12: HTAB 13: PRINT N
840 IN$ =  LEFT$ (IN$ + "

         ",41)
850  IF  LEFT$ (IN$,8) <  > "
         " THEN  GOSUB 1030
860  GOSUB 1200
870  NEXT
880  PRINT D$"CLOSE "PR$
890 LO(NI + 1) = AD
900  GOSUB 1610
910  VTAB 14: CALL  - 958: PRINT
    "SECOND PASS:"
920  FOR N = 2 TO NI
930  VTAB 14: HTAB 13: PRINT N
940  GOSUB 1670
950  NEXT
960  PRINT : PRINT "ASSEMBLY COMP
    LETE": PRINT :L = LO(N - 1) +
    LN(N - 1)
970  IF LA + L < 38400 AND ( PEEK
    (109) +  PEEK (110) * 256) <
    (LA) THEN 1010
980  IF LA + L < 38400 AND LA + L
     > 2049 THEN  PRINT "OBJECT
    CODE MAY OVERLAP APPLESOLF A
    SSEMBLER": GOTO 1010
990  IF LA < 2049 THEN 1010
1000  PRINT "OBJECT CODE  OVERLAP
    S DOS.YOU MUST": PRINT "RE-L
    OCATE IT FROM THE PRESENT LO
    CATION": PRINT "ADDRESS:";SA
    : PRINT "LEGTH:";L: END
1010  PRINT "SAVING ";PR$;".OBJ,A
    ";LA;",L";L: PRINT D$"OPEN B
    .A.TEMP": PRINT D$"CLOSE ": PRINT
    D$"DELETE B.A.TEMP": PRINT D
    $"OPEN B.A.TEMP": PRINT D$"W
    RITE B.A.TEMP ": PRINT D$"BS
    AVE";PR$;".OBJ,A";SA;",L";L
1020  PRINT D$"BLOAD";PR$;".OBJ,A
    ";LA: PRINT D$"BSAVE";PR$;".
    OBJ,A";LA;",L";L: PRINT D$"C
    LOSE": PRINT D$"EXEC B.A.TEM
    P": END
1030  REM
1040  FOR P = 1 TO 8
1050  IF  MID$ (IN$,P,1) = " " THEN
    1070
1060  NEXT
1070 P = P - 1
1080  IF  MID$ (IN$,10,3) <  > "E
    QU" THEN 1160
1090 NF = NF + 1
1100 LF$(NF) =  LEFT$ (IN$,P)
1110 AF(NF) =  VAL ( MID$ (IN$,14
    , LEN (IN$) - 13))
1120 XL =  LEN ( STR$ (AF(NF))) +
    14
1130  IF XL >  =  LEN (IN$) THEN
    1190
1140  IF  MID$ (IN$,XL,1) = "!" THEN
    XA = AF(NF): GOSUB 2420
1150  GOTO 1190
1160 NR = NR + 1
1170 LR$(NR) =  LEFT$ (IN$,P)
1180 DR(NR) = AD + LA
1190  RETURN
1200  REM
1210 P = 14
1220 IN$(N) =  MID$ (IN$,10,3)
1230 LO(N) = AD
1240  IF IN$(N) = "EQU" THEN 1600

1250  IF IN$(N) = "DRS" THEN LN(N
    ) =  VAL ( MID$ (IN$,14)): GOSUB
    2220: GOTO 1590
1260  IF  MID$ (IN$,14,1) = " " THEN
    MD(N) = 1: GOTO 1380
```

```
1270 P = 15
1280  IF  MID$ (IN$,P,1) <  > " "
      AND  MID$ (IN$,P,1) <  > ",
     " AND  MID$ (IN$,P,1) <  > "
     )" THEN P = P + 1: GOTO 1280

1290  IF  MID$ (IN$,P,3) = ",X" THEN
     MD(N) = 5: GOTO 1380
1300  IF  MID$ (IN$,P,3) = ",Y" THEN
     MD(N) = 6: GOTO 1380
1310  IF  MID$ (IN$,P,3) = ",X)" THEN
     MD(N) = 7: GOTO 1380
1320  IF  MID$ (IN$,P,3) = ")" THEN
     MD(N) = 12: GOTO 1380
1330  IF  MID$ (IN$,P,3) = "),Y" THEN
     MD(N) = 8: GOTO 1380
1340  IF  MID$ (IN$,14,1) = "#" THEN
     MD(N) = 4: GOTO 1380
1350  IF  MID$ (IN$,14,1) = "<" THEN
     MD(N) = 13: GOTO 1380
1360  IF  MID$ (IN$,14,1) = ">" THEN
     MD(N) = 14: GOTO 1380
1370 MD(N) = 2
1380  IF  MID$ (IN$,10,1) = "B" AND
      MID$ (IN$,10,3) <  > "BIT" AND
      MID$ (IN$,10,3) <  > "BRK" THEN
     MD(N) = 11:LB$(N) =  MID$ (I
     N$,14,P - 14):TP$(N) = "R": GOTO
     1560
1390  IF MD(N) = 4 THEN LB$(N) =
      MID$ (IN$,15,P - 15):TP$(N)
      = "F": GOTO 1580
1400  IF MD(N) = 13 OR MD(N) = 14
      THEN LB$(N) =  MID$ (IN$,15
     ,P - 15): GOTO 1440
1410  IF MD(N) = 7 OR MD(N) = 8 OR
     MD(N) = 12 THEN LB$(N) =  MID$
     (IN$,15,P - 15): GOTO 1440
1420 LB$(N) =  MID$ (IN$,14,P - 1
     4)
1430  IF MD(N) = 1 THEN TP$(N) =
     "F": GOTO 1560
1440  FOR I = 1 TO NF
1450  IF LF$(I) = LB$(N) THEN TP$
     (N) = "F":LB$(N) =  STR$ (AF
     (I)): GOTO 1520
1460  NEXT
1470  FOR I = 1 TO NR
1480  IF LR$(I) = LB$(N) THEN TP$
     (N) = "D":LB$(N) =  STR$ (DR
     (I)): GOTO 1520
1490  NEXT
1500 TP$(N) = "U": FOR Z = 1 TO N
     F: IF  LEN (LF$(Z)) <  LEN (
     LB$(N)) THEN  IF LF$(Z) =  LEFT$
     (LB$(N), LEN (LF$(Z))) THEN
      IF AF(Z) < 256 THEN 1530
1510  NEXT
1520  IF TP$(N) <  > "F" OR AF(I)
      >  = 256 THEN 1560
1530  IF MD(N) = 2 THEN MD(N) = 3

1540  IF MD(N) = 5 THEN MD(N) = 9

1550  IF MD(N) = 6 THEN MD(N) = 1
     0
1560  IF MD(N) = 1 THEN LN(N) = 1
     : GOTO 1590
1570  IF MD(N) = 2 OR MD(N) = 5 OR
     MD(N) = 6 OR MD(N) = 12 THEN
     LN(N) = 3: GOTO 1590
1580 LN(N) = 2
1590 AD = AD + LN(N)
1600  RETURN
1610  REM
1620  VTAB 13: FLASH : PRINT "
1630  FOR N = 1 TO 56: READ OP$(N
     ): FOR I = 1 TO 12: READ OP(
     N,I): NEXT : NEXT
1640  PRINT
1650  VTAB 13: INVERSE : PRINT "
                  ": NORMAL
1660  RETURN
1670  REM
1680  IF IN$(N) = "DRS" OR IN$(N)
      = "EQU" THEN 2110
1690  FOR I = 1 TO 56
1700  IF OP$(I) = IN$(N) THEN 174
     0
1710  NEXT
1720  PRINT "INVALID INSTRUCTION=
     =>(#"N")"
1730  GOTO 2120
1740  IF (MD(N) = 13 OR MD(N) = 1
     4) AND OP(I,4) <  > 0 THEN 1
     790
1750  IF OP(I,MD(N)) <  > 0 THEN
     1790
1760  IF IN$(N) = "BRK" AND MD(N)
      = 1 THEN 1790
1770  PRINT "INVALID ADDRESS MODE
     ==>(#"N")"
1780  GOTO 2120
1790  IF TP$(N) <  > "U" AND TP$(
     N) <  > "R" THEN 1930
1800  FOR P = 2 TO  LEN (LB$(N))
1810  IF  MID$ (LB$(N),P,1) = "+"
      OR  MID$ (LB$(N),P,1) = "-"
      THEN 1840
1820  NEXT
1830 P = P + 1
1840  IF TP$(N) = "R" THEN 1940
1850  FOR J = 1 TO NF
1860  IF LF$(J) =  LEFT$ (LB$(N),
     P - 1) THEN TP$(N) = "F":LB$
     (N) =  STR$ (AF(J) +  VAL ( MID$
     (LB$(N),P))): GOTO 2000
1870  NEXT
1880  FOR J = 1 TO NR
1890  IF LR$(J) =  LEFT$ (LB$(N),
     P - 1) THEN TP$(N) = "D":LB$
     (N) =  STR$ (DR(J) +  VAL ( MID$
     (LB$(N),P))): GOTO 2000
1900  NEXT
1910  PRINT "INVALID LABEL==>(#"N
     ")"
1920  GOTO 2120
1930  IF TP$(N) <  > "R" THEN 200
     0
1940  FOR J = 1 TO NR
```

APPLESOFT 滙編器

```
1950  IF LR$(J) =  LEFT$ (LB$(N),
      P - 1) THEN LB$(N) =  STR$ (
      DR(J)) +  MID$ (LB$(N),P): GOTO
      1980
1960  NEXT
1970  GOTO 1910
1980 LB$(N) =  STR$ ( VAL (LB$(N)
      ) - (LO(N + .1) + LA))
1990  IF  VAL (LB$(N)) > 127 OR  VAL
      (LB$(N)) <  - 128 THEN  PRINT
      "INVALID RELATIVE ADDRESS==>
      (#"N")": PRINT "="LA$(N): GOTO
      2120
2000  IF MD(N) = 13 THEN  POKE LO
      (N) + SA,OP(I,4):LB$(N) =  STR$
      ( INT ( VAL (LB$(N)) / 256))
      : GOTO 2040
2010  IF MD(N) = 14 THEN  POKE LO
      (N) + SA,OP(I,4):LB$(N) =  STR$
      ( VAL (LB$(N)) - (256 *  INT
      ( VAL (LB$(N)) / 256))): GOTO
      2040
2020  POKE LO(N) + SA,OP(I,MD(N))

2030  IF LN(N) = 1 THEN 2110
2040  IF LN(N) = 2 AND  VAL (LB$(
      N)) > 255 THEN  PRINT "INVAL
      ID LENGHT,>255==>(#"N")": GOTO
      2120
2050  IF  VAL (LB$(N)) >  = 0 THEN
      2080
2060 LB = 256 +  VAL (LB$(N))
2070 LB$(N) =  STR$ (LB)
2080  IF LN(N) = 2 THEN  POKE LO(
      N) + SA + 1, VAL (LB$(N)): GOTO
      2110
2090  POKE LO(N) + SA + 2, VAL (L
      B$(N)) / 256
2100  POKE LO(N) + SA + 1, VAL (L
      B$(N)) - (256 *  PEEK (LO(N)
      + SA + 2))
2110  RETURN
2120  REM
2130  PRINT D$"OPEN"PR$: PRINT D$
      "READ "PR$
2140  FOR X = 1 TO N
2150  INPUT IN$
2160  NEXT
2170  PRINT D$"CLOSE "PR$
2180  PRINT IN$
2190  VTAB 22: CALL  - 958: INPUT
      "DO YOU WISH TO EDIT THE FIL
      E?(Y/N)";K$: IF K$ = "Y" THEN
       PRINT D$"RUN ASSEMBLER.EDIT
      OR"
2200  IF K$ = "N" THEN  HOME : END

2210  PRINT  CHR$ (7): GOTO 2190
2220  REM
2230 XL =  LEN ( STR$ (LN(N))) +
      14
2240  IF XL >  =  LEN (IN$) THEN
       RETURN
2250  IF  MID$ (IN$,XL,1) = "!" THEN
      XA = LO(N) + SA: GOSUB 2420
2260  RETURN
2270  DIM PN(16)
2280 PN(1) = 67:PN(2) = 79:PN(3) =
      77:PN(4) = 80
2290 PN(5) = 85:PN(6) = 84:PN(7) =
      73:PN(8) = 78
2300 PN(9) = 71:PN(10) = 32:PN(11
      ) = 45:PN(12) = 32
2310 PN(13) = 65:PN(14) = 71:PN(1
      5) = 69:PN(16) = 32
2330  FOR VN = 1 TO 16
2340  VTAB 5
2350  FOR VO = VN TO 1 STEP  - 1
2360  PRINT  CHR$ (PN(17 - VO));
2370  NEXT VO
2380  PRINT
2390  NEXT VN
2400  PRINT
2410  RETURN
2420  REM
2430 XL = XL + 1
2440  IF XL >  LEN (IN$) THEN  RETURN

2450  IF  MID$ (IN$,XL,1) = "!" THEN
       RETURN
2460  POKE XA, ASC ( MID$ (IN$,XL
      ,1))
2470 XA = XA + 1
2480  GOTO 2420
2490 HF = 0:DT = 0
2500  FOR K = 1 TO  LEN (HX$)
2510  IF  MID$ (HX$,K,1) = "$" THEN
      HV$ =  MID$ (HX$,K): IF  LEN
      (HV$) > 1 THEN HV$ =  MID$ (
      HV$,2): GOTO 2530
2520  NEXT K: RETURN
2530  IF  LEN (HV$) < 4 THEN  PRINT
      D$"CLOSE": PRINT "OPERAND ER
      ROR IN LINE:";N: GOTO 2180
2540  FOR K1 = 1 TO 4
2550 HD =  ASC ( MID$ (HV$,K1,1))

2560 HD = HD - 48: IF HD > 9 THEN
      HD = HD - 7: IF HD >  = 10 AND
      HD <  = 15 THEN 2590
2570  IF HD > = 0 AND HD <  = 9 THEN
      2590
2580  PRINT D$"CLOSE": PRINT "OPE
      RAND ERROR IN LINE:";N: GOTO
      2180
2590 DT = DT + HD * (16 ^ (4 - K1
      ))
2600  NEXT
2610  IF K > 1 THEN HX$ =  LEFT$
      (HX$,K - 1) +  STR$ (DT) +  MID$
      (HX$,K + 5): GOTO 2630
2620 HX$ =  STR$ (DT) +  MID$ (HX
      $,6)
2630  RETURN
```

眞正要動腦筋的小學四則運算

本欄過去曾刊登過幾個小學加減乘除的運算程式，幫助大家同學練習運算四則算術。通常，這些程式在執行時，會先顯示問題，然後要求孩子打入答案。部份家長可能會認爲此類程式可能會導致孩子們只懂得按掣尋求答案，而忽略了四則運算基礎，一如昔日我們爭辯中學生應否使用計數機一樣。

不過，各位家長不用再受這類問題困擾，我們今期介紹的一個小學四則運算程式，是專針對此點而設計，程式完全參照實際課堂上的教導方法。孩子們必須要眞正的去動腦筋才能計算出答案，而且計算過程有如用筆算一樣。

舉個例子，在除數運算模式中，孩子們必須每一個數都要計算，一步跟一步，而非再是單純輸入答案便可，然後等待電腦告訴答案是或否。

例如：1776÷16

計算步驟如下：

```
       111
16 / 1776
     16       ——— (1)
      17      ——— (2)
      16      ——— (3)
       16     ——— (4)
       16     ——— (5)
        0     ——— (6)
```

孩子們必須將（1）至（6）每一個步驟計算清楚，電腦是不會自動計算的。如此做法，孩子們必需要練習心算。

程式開始執行時，會有一系列的指示指導如何使用本程式，家長或兄長們最好是能先了解一次然後解釋給孩子知道。程式中有多個按鍵是用來協助在運算過程中控制各項程序。要觀看這些〔按鍵〕，可直接按入〔？〕。在每個運算模式中都會同時顯示出這些指令鍵的撮要表。

程式執行時，電腦會要求孩子輸入自己的名字，然後是選擇運算模式，同時亦要決定每條題目中每個數的數位數目。在除算模式中，孩童還要決定被除數是否有餘數。

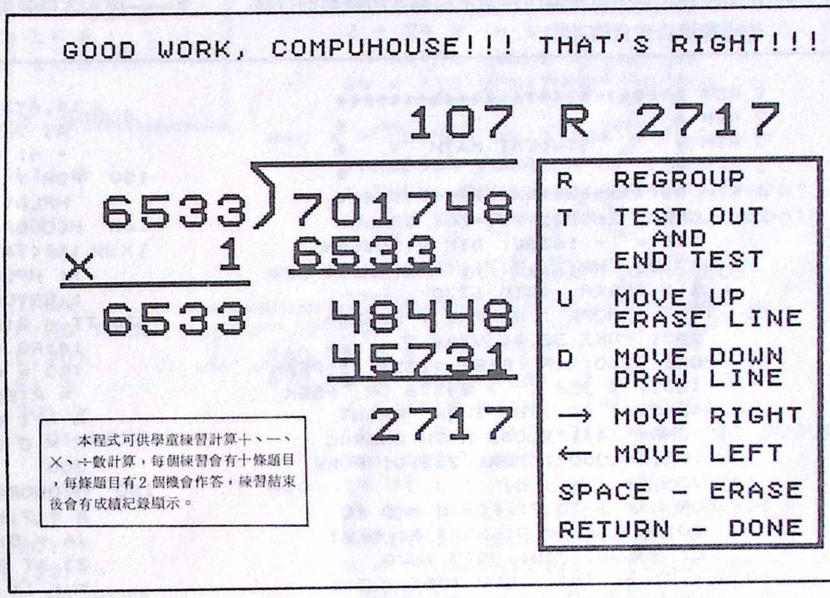
加數及減數問題，上限數最多不能多個8位數位，下限數最多不能多過6位數字。乘數問題亦一樣。至於除數算式中，被除數可以長至6個數位，而除數則不能超過4個數位，或者在任何情況之下被除數不能長過除數超過二個數位。

在乘數中的答案，最長只能有14個數位。由於顯示答案的長度是受到限制，故此乘數最多只能有6個數位。

在除算模式中，垂直方向最多只能有三行，水平方向最多只能有20個字位。除數佔4個，被除數佔6個，除算符號佔1個，指示出餘數的R（EMAINDER）佔一個。

決定完計算問題方式後，電腦會詢問是否有輸入錯誤，如無誤，電腦會問你是否要熟讀各個控制鍵說明。表（1）是所有的控制鍵總表。

R鍵是用來將游標移至進位備忘位置，供孩子們在計算進位時所用。在加算和減算之中，通常是在問題之上的位置。左右箭咀鍵是用來移動游標。



本程式可供學童練習計算十、一、×、÷數計算，每個練習會有十條題目，每條題目有2個機會作答，練習結束後會有成績紀錄顯示。

在乘及除算中，有兩個位置供進位計算用。一是在問題之項上，另一是在問題答案部份。在乘算中，是用來計算最後的結果，在除算中是用來消減計算。

SPACE BAR 是用來擦去游標所處位置上之數字。假如位置是空白，游標會右移。

假如要改變數目，按入新數目卽可。當使用L綸劃行綫時（在乘算模式），游標會跳落下一行。T只用於除算模式。它將游標移至除數之下及劃出乘號或行線。孩子們可用來作草算。當完成草算後，再按T會叫游標返回原先位置。進位數位置及測試部份會全部留下直至再按 TEST OUT（T鍵）爲止。

本教育程式共分兩個 FILE，它們包括：（1）STUDENT MATH，（2）MATH CHARACTER。□

眞正要動腦筋的小學四則運算

表（1）

R——將游標移至要進位的位置。
U——將游標向上移，在除數模式用來消除未用的間線。
箭咀——將游標右左移動（不會刪除任何東西）
空位鍵——刪除游標上之數目或任何進位數（R）
L——在乘數模式中，用來劃下或取消行線。所繪劃的行線是跟游標所處位置相同。
T——在除數模式中，設立草算或返回原先執行的步驟。
RETURN——當完成計算後，告訴電腦，電腦會檢查答案是對或錯。
ESC ——停止計算問題，重新選擇計算模式。
?——顯示本控制鍵列表。

```
1 REM **************************
2 REM *                        *
3 REM *      STUDENT MATH      *
5 REM *                        *
6 REM **************************
20  LOMEM: 26000: ROT= 0: SCALE=
    1:P =  - 16368: DIM R(20),AN
    (7,20),NR(2),E(7),TN(8),N$(1
    1): ONERR  GOTO 1770
30  TEXT : HOME : NORMAL : SPEED=
    255: POKE 32,9: VTAB 7
40  POKE 32,0: IF  PEEK (232) +  PEEK
    (233) * 256 <  > 24576 OR  PEEK
    (24576) <  > 82 THEN  PRINT
     CHR$ (4);"BLOAD MATH CHARAC
    TER,A$6000": POKE 232,0: POKE
    233,96
50  FOR L = 1 TO 7: FOR E = 0 TO
    20:AN(L,E) = 0: NEXT E: NEXT
    L: GOSUB 1760: GOTO 1070
60 S = 21 -  INT ( LEN (S$) / 2):
     RETURN
70 BX = X: FOR T = 1 TO  LEN (S$)
    :A$ =  MID$ (S$,T,1): IF  ASC
    (A$) <  > 32 THEN  XDRAW  ASC
    (A$) - 26 + 48 * AN AT BX,Y
80 BX = BX + 7 + 7 * AN: NEXT : RETURN

90 S$ = N$ + ", THIS IS TRY " +  STR$
    (TRY) + " ON PROBLEM " +  STR$
    (W) + ".": GOSUB 60:X = (S -
    1) * 7:Y = AY(0) - 8 * (A =
    4): GOSUB 70: IF W = 1 THEN
     RETURN
100 S$ = "SO FAR YOU HAVE " +  STR$
    (NR(1) + NR(2)) + " RIGHT OU
    T OF " +  STR$ (W - 1) + "."
    : GOSUB 60:X = (S - 1) * 7:Y
     = Y + 16: GOSUB 70: RETURN

110 TT = UL + L * (UL = 0): FOR T
     = AX(TT) TO AX STEP 14: XDRAW
    1 AT T,AY(TT): NEXT T:UL = L
     * (UL = 0): IF R(20) = 0 THEN
    SR = 0:R = 0:RG = 0: RETURN

120 RG = 0: FOR EC = 19 TO ES STEP
     - 1:RT = R(EC) -  INT (R(EC
    ) / 100) * 100: IF RT THEN S
    U = 0: GOSUB 170: XDRAW RT +
    60 - 48 * R AT RX,Y:R(20) =
    R(20) - R(EC):R(EC) = 0:RG =
    0: IF R(20) = 0 THEN EC = ES

130  NEXT EC:SR = 0:R = 0: RETURN

140  FOR T = E(2) + 2 TO E(2) + 2
     + BD: XDRAW 1 AT 266 - T *
    14,AY(2) + 4: NEXT : XDRAW 5
     AT 266 - (T - 1) * 14,AY(2)
     + 4: RETURN
150  FOR Y = AY(0) TO AY(0) + 15:
     HPLOT 0,Y TO 62,Y: NEXT Y
160  HCOLOR= 3: FOR Y = 152 + 24 *
    (A < 4) TO 167 + 24 * (A < 4
    ): HPLOT 0,Y TO 76,Y: NEXT Y
    : RETURN
170 TT = 0:RX = 266 - (EC + SR) *
    14:RG =  INT (R(EC) / 1000) *
    (RG = 0) + RG:Y = RG - 100:T
     = R(EC) - RG * 1000 - RT: IF
    A = 1 OR A = 4 OR A = 5 AND
    R = 0 OR T = 0 AND SU = 0 THEN
    200
180  HCOLOR= 3 * SU: DRAW 80 AT R
    X - 7 * (A = 5),Y + 16 - 8 *
    (A = 5) + 8 * (A = 2): DRAW
    23 AT RX + 14 - 5 * (A = 5),
    Y + 16 * (A = 2):TT = 1: IF
    R(EC + 1) > 0 AND SU OR T =
    200 THEN  DRAW 66 AT RX - 6 *
    (A = 5),Y + 16 * (A = 2):TT =
    2
190  IF SU = 0 THEN  HCOLOR= 3: DRAW
    TN(EC - ES + 1) * (L < 4) +
    AN(L - 2 * (L > 3),EC + 1) *
    (L > 3) AT RX - 7 * (A = 5),
    Y + 16 - 8 * (A = 5) + 8 * (
    A = 2): IF R(EC + 1) > 0 THEN
     DRAW 23 AT RX - 5 * (A = 5)
    ,Y + 16 * (A = 2)
200 R(EC) = RG * 1000 + TT * 100 +
    RT:SU = 0: RETURN
210 UL = 0:LL = L - (K = 68): FOR
    T = 7 TO LL + 1 STEP  - 1:UL
     = UL + AN(T,20): NEXT T: FOR
    T = 14 * (LL = 2) TO 19 * (L
    L = 2):UL = UL - AN(3,T): NEXT
    T: IF UL <  > 0 OR AN(LL,20)
     = 0 THEN  NEXT Z
220  FOR T = ES TO LE: IF AN(LL,T
    ) = 0 AND AN(LL - 1,T) * (LL
     > 2) = 0 THEN  NEXT T
230  FOR S = (LE) TO ES STEP  - 1
    : IF AN(LL,S) = 0 AND AN(LL -
    1,S) = 0 THEN  NEXT S
240 S = S * (LL > 2) + LE * (LL =
    2): IF S < T THEN  NEXT Z
250  FOR SS = T TO S: IF ((AN(LL,
    SS) < 100) + (K = 85)) = 1 THEN
     XDRAW 1 AT AX - (SS - ES) *
    14,AY(LL):AN(LL,SS) = AN(LL,
    SS) + 100 - 200 * (K = 85):A
    N(LL,20) = AN(LL,20) + 100 -
    200 * (K = 85)
260  NEXT SS: NEXT Z
```

眞正要動腦筋的小學四則運算

```
270  IF R(20) > 0 THEN  GOSUB 120

280  FOR L = 7 TO 1 STEP  - 1: IF
     AN(L,20) = 0 AND ((UL = L) *
     (A = 4)) = 0) THEN  NEXT L: GOTO
     340
290  IF UL = L AND A = 4 THEN  GOSUB
     110:UL = 0
300  FOR E = E(L) + (19 - E(L)) *
     (A = 5) * (L < 4) * (L <  >
     1) TO 0 STEP  - 1:SH = AN(L,
     E): IF SH = 0 THEN  NEXT E: NEXT
     L: GOTO 340
310 X = 266 - E * 14: IF SH THEN
     HS =  INT (SH / 100): IF HS THEN
      XDRAW 1 AT X,AY(L):AN(L,20)
      = AN(L,20) - 100:AN(L,E) =
     AN(L,E) - 100: IF AN(L,20) =
     0 AND R(20) = 0 THEN E = 0: NEXT
     E: NEXT L: GOTO 340
320  IF SH THEN SH = SH - HS * 10
     0: IF SH THEN  XDRAW SH AT X
     ,AY(L):AN(L,20) = AN(L,20) -
     SH:AN(L,E) = 0: IF AN(L,20) =
     0 AND R(20) = 0 THEN E = 0: NEXT
     E: NEXT L: GOTO 340
330  NEXT E
340  IF TS THEN  GOSUB 140
350  RETURN
360  GOSUB 90:LL = L:L = 1:AN = 0
     :TRY = TRY + 1: IF A = 4 THEN
      FOR T = 7 TO 1 STEP  - 1:L =
     T:T = T - (T - 1) * (AN(L,20
     ) > 0): NEXT T: IF AN(L,20) =
     0 THEN 410
370  FOR T = 19 TO ES STEP  - 1:E
      = T:T = T - (T - ES) * (AN(
     L,E) > 0): NEXT T:AN$ = "": FOR
     T = ES TO E:S = AN(L,T): IF
     S < 70 OR S > 79 THEN T = E:
      NEXT T: GOTO 410
380 AN$ =  STR$ (S - 70) + AN$: NEXT
     T: IF A = 5 AND R$ = "Y" THEN
      FOR S = 0 TO ES - 4:E = S:S
      = S + ES * (AN(1,E) > 0): NEXT
     S: IF S = ES - 3 THEN 410
390  IF A = 5 AND R$ = "Y" THEN R
     M = 0:TT = 0: FOR T = E TO E
     S - 4:S = AN(1,T):RM = RM +
     (S - 70) * (S > 0) * 10 ^ TT
     :TT = TT + 1: NEXT T:AN$ = A
     N$ + " " +  CHR$ (47) + " " +
      STR$ (RM)
400  IF CA$ = AN$ THEN L = LL: GOTO
     460
410 TT = 0:L = LL: IF TRY = 2 THEN
     500
420  HCOLOR= 3: GOSUB 150
430 X = 0:Y = AY(0):S$ = "THE RIG
     HT": GOSUB 70:Y = Y + 8:S$ =
     "ANSWER IS": GOSUB 70:Y = 15
     2 + 24 * (A < 4):S$ = "PRESS
      A KEY": GOSUB 70:Y = Y + 8:
     S$ = "TO CONTINUE": GOSUB 70

440 X = LX:Y = AY(0):S$ = CA$:AN =
     1: GOSUB 70:AN = 0: IF TT <
     1 THEN  POKE P,0: GET A$: PRINT
     :K =  ASC (A$):TT = TT + 1: GOTO
     440
450  HCOLOR= 0: GOSUB 150: GOSUB
     270:W = W + (K = 27): GOTO 6
     10
460 TT = 0
470 S$ = "GOOD WORK, " + N$ + "!!
     ! THAT'S RIGHT!!!": GOSUB 60
     :X = (S - 1) * 7:Y = AY(0): GOSUB
     70: GOSUB 160
480  IF TT < 1 THEN X = 0:Y = 152
      + 24 * (A < 4):S$ = "PRESS
     A KEY": GOSUB 70:Y = Y + 8:S
     $ = "TO CONTINUE": GOSUB 70:
      POKE P,0: GET A$: PRINT :K =
      ASC (A$):TT = 2: GOTO 470
490 NR(TRY - 1) = NR(TRY - 1) + 1
     : GOSUB 270:W = W + (K = 27)
     : GOTO 610
500 TT = 0
510 S$ = "NOPE, THAT'S NOT RIGHT,
      " + N$ + ".": GOSUB 60:X =
     (S - 1) * 7:Y = AY(0) - 8 *
     (A = 4): GOSUB 70:S$ = "TRY
     AGAIN!": GOSUB 60:X = (S - 1
     ) * 7:Y = Y + 16: GOSUB 70: GOSUB
     160
520  IF TT < 1 THEN X = 0:Y = 152
      + 24 * (A < 4):S$ = "PRESS
     A KEY": GOSUB 70:Y = Y + 8:S
     $ = "TO CONTINUE": GOSUB 70:
      POKE P,0: GET A$: PRINT :K =
      ASC (A$):TT = 2: GOTO 510
530  GOSUB 270: ON ((K = 27) + 1)
      GOTO 700,610
540 ES = (266 - AX) / 14:K = 0:NR
     (1) = 0:NR(2) = 0: GOSUB 160
     : FOR W = 1 TO 10: FOR T = 0
      TO 8:TN(T) = 0: NEXT T:TRY =
     1: IF AA = 3 THEN A =  INT (
      RND (1) * 2) + 1
550  IF K = 27 THEN  GOSUB 160: GOTO
     1590
560 AX(1) = SX - (A = 2) * 14:E(1
     ) = LE - (A = 2):RY = YY - 8
      * (A = 2):TN =  INT ( RND (
     1) * 10 ^ TD + 1): IF  LEN (
      STR$ (TN)) <  > TD THEN 560
```

眞正要動腦筋的小學四則運算

```
570 BN =  INT ( RND (1) * 10 ^ BD
    ) + (A = 5): IF  LEN ( STR$
    (BN)) <  > BD THEN 570
580  IF BN > TN THEN T = BN:BN =
    TN:TN = T:T = 0
590  IF A = 5 AND R$ = "Y" THEN  IF
    TN / BN =  INT (TN / BN) THEN
    560
600  IF A = 5 AND R$ <  > "Y" THEN
    TN =  INT (TN / BN) * BN: IF
    BN = TN OR  LEN ( STR$ (TN))
     < TD THEN 560
610 TT = 0: FOR T = 0 TO TD - 1:X
     = AX - 14 * TT:Y = BY - 24 *
    (A = 5):TN(T) =  VAL ( MID$
    ( STR$ (TN),TD - T,1)) + 70:
     XDRAW TN(T) AT X,TY:TT = TT
     + 1: XDRAW 1 AT X,Y: NEXT T

620  FOR T = 1 TO 1 + (A < 5):X =
    X - 14: XDRAW 1 AT X,Y: NEXT
    T: XDRAW A + 1 + (A < 4) AT
    X,BY:TT = (TT + 1) * (A = 5)

630  FOR T = 0 TO BD - 1: XDRAW  VAL
    ( MID$ ( STR$ (BN),BD - T,1)
    ) + 70 AT AX - 14 * TT,BY:TT
     = TT + 1: NEXT T: IF A = 5 AND
    R$ = "Y" THEN  XDRAW 69 AT A
    X + 28,AY(1)
640  IF TRY > 1 THEN  NEXT W: GOTO
    1590
650 CA$ = "": IF A = 1 THEN CA$ =
     STR$ (TN + BN)
660  IF A = 2 THEN CA$ =  STR$ (T
    N - BN)
670  IF A = 4 THEN TT = 0: FOR T =
    1 TO BD:S = TN *  VAL ( MID$
    ( STR$ (BN),BD - T + 1,1)) +
    TT:CA$ =  STR$ (S -  INT (S /
    10) * 10) + CA$:TT =  INT (S
     / 10): NEXT T: IF TT THEN C
    A$ =  STR$ (TT) + CA$
680  IF A = 5 THEN S =  INT (TN /
    BN):CA$ =  STR$ (S)
690 LX = AX - 14 * ( LEN (CA$) -
    1): IF A = 5 AND R$ = "Y" THEN
    CA$ = CA$ + " " +  CHR$ (47)
     + " " +  STR$ (TN - BN * S)

700  GOSUB 90:X = 7:Y = 152 + 24 *
    (A < 4):S$ = "PRESS  ?": GOSUB
    70:Y = Y + 8:S$ = "FOR  HELP
    ": GOSUB 70:L = 1:CR = 0:R =
    0:TS = 0:RG = 0:RE = 0:SR =
    0:E = ES:UL = 0:LL = 0:SH =
    0:SU = 0
710  POKE P,0: FOR Z = 1 TO 2:Z =
    1:RE = RE * (RG > 0): IF A =
    2 OR A = 5 AND R = 1 THEN RE
     = RE + ((E - RE) > E(L)) *
    (RG > 0)
720 EC = E - RE - SR * (RE <  > 0
    ):X = 266 - (E - RE) * 14:Y =
    AY(L) * (RG = 0) + (RG - 100
    ) * (RG > 0)
730  FOR RC = 1 TO 2:CR = (CR = 0
    ): XDRAW 1 + R * (RG > 0) AT
    X,Y: FOR C = 1 TO 13:K =  RND
    (1):K =  PEEK ( - 16384):C =
    C + 15 * (K > 128):RC = 2 *
    (K > 128): NEXT C: NEXT RC: IF
    CR THEN  XDRAW 1 + R * (RG >
    0) AT X,Y
740  POKE P,0:CR = 0:C = 0:K = K -
    128: IF K = 32 OR K > 47 AND
    K < 58 THEN C = 1:SH = ((K +
    22) - 48 * R * (RG > 0)) * (
    K <  > 32):HS = AN(L,E) -  INT
    (AN(L,E) / 100) * 100
750  IF C AND HS AND RG = 0 THEN
     XDRAW HS AT X,Y:AN(L,20) =
    AN(L,20) - HS:AN(L,E) = AN(L
    ,E) - HS
760  IF C THEN RT = R(EC) -  INT
    (R(EC) / 100) * 100: IF (RT)
     AND (RG OR (SH = 0 AND HS =
    0)) THEN SU = 0:HR = RG: GOSUB
    170: XDRAW RT + 60 - 48 * R AT
    RX,Y:R(20) = R(20) - R(EC):R
    (EC) = 0:RG = HR
770  IF C AND SH THEN  XDRAW SH AT
    X,Y:TT = SH * (RG = 0):AN(L,
    E) = AN(L,E) + TT:AN(L,20) =
    AN(L,20) + TT: IF RG THEN RT
     = (SH - 60 + 48 * R):R(EC) =
    RG * 1000 + RT:SU = 1: GOSUB
    170:R(20) = R(20) + R(EC)
780  IF C THEN TT = (SH > 0) * ((
    E <  > E(L) + (BD + 2) * (TS
     = 2)) - (E < ES) * (E > ES -
    7) - (A = 5) * (L = 1 + (TS =
    2))) * (E <  > E(L))
790  IF C THEN TT = TT - (SH = 0)
     * ((RG = 0) * (E <  > ES) *
    (E <  > E(L) + 2) * (E > ES -
    7) * (RT = 0) - (RG > 0))
800  IF C THEN E = E + TT * (RG =
    0):RE = RE + TT:R = R * (R(2
    0) > 0):SR = SR * (R(20) > 0
    ):RG = RG * (RE < 0) * (((A =
    5) AND R) OR (A = 2)): NEXT
    Z
810  IF K = 63 THEN  TEXT : HOME
    : GOSUB 1640: POKE  - 16299,
    0: POKE  - 16304,0: NEXT Z
```

眞正要動腦筋的小學四則運算

```
820  IF K = 27 THEN Z = 2: NEXT Z
     : GOSUB 90: GOSUB 270:TRY =
     2: GOTO 610
830  IF K = 13 THEN Z = 2: NEXT Z
     : GOTO 360
840  IF K = 8 THEN C = 1:T = (R =
     0) * (RG > 0) * (A = 4): IF
     A = 5 AND RG > 0 AND R = 1 AND
     L > 3 AND AN(L - 2 * (L > 3)
     ,EC + 2) = 0 THEN  NEXT Z
850  IF C AND EC < E(L * (T = 0) +
     (T > 0)) + (BD + 2) * (A = 5
     ) * (L > 1 + (TS = 2)) * ((R
     G > 0) * (R = 0) OR (TS = 2)
     ) THEN E = E + (RG = 0):RE =
     RE - 1: IF E = ES - 3 THEN E
      = E + 3
860  IF C THEN  NEXT Z
870  IF K = 21 THEN C = 1: IF EC >
     ES + (TD + 1) * (A = 5) * (R
      = 0) * ((TS = 2) OR (RG > 0
     )) THEN E = E - (RG = 0):RE =
     RE + 1: NEXT Z
880  IF C AND A = 5 AND L = 1 AND
     R$ = "Y" AND E > ES - 7 THEN
     E = E - 1: IF E = ES - 1 THEN
     E = E - 3
890  IF C THEN  NEXT Z
900  IF K = 85 AND L > 1 + (TS =
     2) AND RG = 0 THEN  ON (R(20
     ) > 0) GOSUB 120:L = L - 1:T
      = E(L) + 2 * (TS = 2):E = E
      * (E <  = T) + T * (E > T):
      ON ((A = 5) * (TS <  > 2) *
     (L / 2 =  INT (L / 2))) GOTO
     210: NEXT Z
910  IF K = 85 THEN  NEXT Z
920  IF K = 68 THEN C = 1: IF RG =
     0 AND (A < 4 OR L > 6 OR TS =
     2 AND L = 3) OR E < ES THEN
      NEXT Z
930  IF C THEN  ON ((RG = 0) * R(
     20) > 0) GOSUB 120:L = L + (
     RG = 0):RG = 0: ON ((A = 5) *
     (TS <  > 2) * (L / 2 <  >  INT
     (L / 2))) GOTO 210: NEXT Z
940  IF C THEN  NEXT Z
950  IF K = 76 AND A = 4 AND L <
     7 + (UL > 0) THEN  GOSUB 110
     :L = L + (UL > 0): NEXT Z
960  IF K = 84 AND A = 5 THEN C =
     1: GOSUB 120: IF TS = 2 THEN
     L = LL:E = EE:TS = 1: NEXT Z

970  IF C THEN LL = L:L = 2:EE =
     E:E = E(2) + 2: FOR TT = 2 TO
     3: FOR T = 19 TO 14 STEP  -
     1:SH = AN(TT,T): IF SH THEN
      XDRAW SH AT 266 - T * 14,AY
     (TT):AN(TT,20) = AN(TT,20) -
     SH:AN(TT,T) = 0
980  IF C THEN  NEXT T: NEXT TT: ON
     (TS = 0) GOSUB 140:TS = 2: NEXT
     Z
990  IF K <  > 82 OR RG OR A = 2 AND
     EC = E(1) OR A = 5 AND (L =
     1 + (TS = 2) OR TS <  > 2 AND
     L / 2 <  >  INT (L / 2) AND
     (AN(L - 1,E) + AN(L - 1,E +
     1) < 200 OR AN(L - 2 * (L >
     1),E + 1) = 0 AND L > 3)) THEN
      NEXT Z
1000 RG = RY:R = 0: IF A = 5 AND
     L > 1 AND L / 2 <  >  INT (L
      / 2) AND TS <  > 2 THEN RG =
     AY(L) + 52:SR = .5:RE =  - .
     5:R = 1: NEXT Z
1010  IF A = 4 AND UL AND L > UL THEN
      FOR S = 1 TO L:T = S:S = S +
     6 * (AN(T,E) > 0): NEXT S:RG
      = AY(T) + 92 - (T = L):R =
     1:RE = .25:SR =  - .25: NEXT
     Z
1020  IF A > 3 THEN  FOR S = ES TO
     E:T = S:S = S + 20 * (AN(L,S
     ) <  > 0 AND AN(L,T) <  > 10
     0): NEXT S
1030  IF A = 4 THEN RE = (T - ES)
      * (R = 0):RE = RE * (E - RE
      <  = LE) + (E - LE) * (E -
     RE > LE)
1040  IF A = 5 THEN RE = ( - 1) *
     (TD + 1 - T + ES) * (R = 0) *
     (TS < 2):RE = RE * (E - RE <
     20) + (E - 19) * (E - RE > 1
     9)
1050  IF A = 2 THEN SR = 1:RE =  -
     1
1060  NEXT Z
1070  HOME : VTAB 10: POKE P,0: PRINT
      TAB( 3)"DO YOU WANT INSTRUC
     TIONS? (Y/N) > ";: GET A$: PRINT
     A$: IF A$ <  > "Y" THEN 1170

1080  HOME : HTAB 14: INVERSE : PRINT
     " INSTRUCTIONS ": NORMAL : POKE
     34,2: PRINT : PRINT "      'S
     TUDENT MATH.'  CAN HELP YOU
     WITH YOUR ADDING,  SUBTRACTI
     NG,  MULTIPLYING AND DIVIDIN
     G SKILLS."
1090  PRINT : PRINT "      I'LL AS
     K YOU TO CHOOSE THE KIND OF
     PROBLEM YOU WANT TO WORK ON
     AND HOW BIG YOU WANT IT TO B
     E."
```

眞正要動腦筋的小學四則運算

```
1100  PRINT : PRINT "     I'LL GI
      VE YOU TEN PROBLEMS. YOU'LL
      HAVE TWO CHANCES TO ANSWER E
      ACH ONE."
1110  PRINT : PRINT "     I'LL KE
      EP SCORE AND SHOW IT TO YOU
      WHILE YOU'RE WORKING ON PROB
      LEMS.  WHEN YOU'RE DONE YOU'
      LL SEE THE FINAL SCORE. THEN
      , IF YOU WANT, YOU CAN TRY T
      EN MORE PROBLEMS, EITHER THE
       SAME KIND OR SOME- THING DI
      FFERENT."
1120  GOSUB 1760
1130  PRINT "     IF YOU WANT TO
      STOP BEFORE ALL TEN PROBLEMS
       ARE DONE, PRESS THE <ESC> K
      EY."
1140  PRINT : PRINT "     A SHORT
       LIST OF THE  KEY FUNCTIONS
      WILL BE DISPLAYED WITH THE P
      ROBLEMS.  I ALSO HAVE A MORE
       DETAILED LIST THAT YOU CAN
      SEE BEFORE STARTING OR WHILE
       YOU'RE"
1150  PRINT "WORKING. TO SEE IT W
      HILE YOU'RE WORKING JUST PRE
      SS THE <?> KEY."
1160  POKE 34,0: GOSUB 1760
1170  FOR T = 1 TO 10:N$(T) = "":
       NEXT T: HOME : INVERSE : HTAB
      13: PRINT " STUDENT MATH. ":
       NORMAL : VTAB 23: PRINT  TAB(
      4)"WHEN YOU'RE DONE, PRESS <
      RETURN>."
1180 N$ = "": POKE P,0: VTAB 10: PRINT
       TAB( 5)"WHAT'S YOUR NAME?
        .........": VTAB 10: HTAB
      26: FOR T = 0 TO 10: GET A$:
      A =  PEEK ( - 16384): IF A >
      31 AND T < 10 THEN  PRINT A$
      ;:N$(T + 1) = A$
1190  IF A = 8 AND T > 0 THEN  PRINT
      A$;:T = T - 1
1200  IF A = 21 THEN A = 1: IF N$
      (T + 1) <  > "" THEN A = 21:
       PRINT N$(T + 1);
1210 S = T:T = T - (A < 32) * (A <
       > 21) - (A > 31) * (T = 10)
       + 20 * (A = 13): NEXT T: FOR
      A = 1 TO S + (T < 19):N$ = N
      $ + N$(A): NEXT A:AA = 6: IF
      S = 0 THEN N$ = ""
1220  HOME : VTAB 10: HTAB 14: PRINT
      "YOUR NAME IS ": PRINT :S$ =
      N$: GOSUB 60: HTAB S: PRINT
      S$: PRINT : PRINT  TAB( 9)"I
      S THAT RIGHT? (Y/N) > ";: GET
      A$: PRINT A$: IF A$ <  > "Y"
       THEN 1170
1230  POKE P,0: HOME :S$ = "HELLO
      , " + N$ + ", WELCOME TO": GOSUB
      60: PRINT  TAB( S)S$: PRINT
      : PRINT : PRINT : HTAB 13: INVERSE :
       PRINT " STUDENT MATH. ": NORMAL
      : PRINT : PRINT : PRINT : PRINT
      "WE HAVE        1. ADDITION"
1240  PRINT : PRINT  TAB( 16)"2.
      SUBTRACTION": PRINT : PRINT
       TAB( 16)"3. BOTH": PRINT : PRINT
       TAB( 16)"4. MULTIPLICATION"
      : PRINT : PRINT  TAB( 16)"5.
       LONG DIVISION": PRINT : PRINT
      : PRINT : PRINT "YOUR CHOICE
      ? > ";
1250  FOR T = 0 TO 7:AX(T) = 0:AY
      (T) = 0:E(T) = 0: NEXT :RY =
      0:TY = 0:BY = 0:AX = 0:TN$ =
      "":BN$ = "":TD = 0:BD = 0:TN
       = 0:BN = 0:BM = 0:AN = 0
1260  GET A$:A =  PEEK ( - 16384)
       - 48: IF A < 1 OR A > 5 THEN
      1260
1270  PRINT A$: HOME : VTAB 7: INVERSE

1280  IF A = 1 THEN S$ = " ADDITI
      ON "
1290  IF A = 2 THEN S$ = " SUBTRA
      CTION "
1300  IF A = 3 THEN S$ = " ADDITI
      ON AND SUBTRACTION "
1310  IF A = 4 THEN S$ = " MULTIP
      LICATION "
1320  IF A = 5 THEN S$ = " LONG D
      IVISION "
1330  GOSUB 60: HTAB S: PRINT S$:
       NORMAL : PRINT : PRINT : PRINT
      :S$ = "SELECT THE NUMBER OF
      DIGITS": GOSUB 60: HTAB S: PRINT
      S$: PRINT : PRINT : IF A < 5
       THEN  PRINT "TOP NUMBER 1 T
      O 8 DIGITS > ";
1340  IF A = 5 THEN  PRINT "DIVIS
      OR 1 TO 4 DIGITS > ";
1350  POKE P,0: GET A$:TD =  PEEK
      ( - 16384) - 48: IF TD < 1 OR
      TD > 8 - 4 * (A = 5) THEN 13
      50
1360 BM = TD * (A < 4) + TD * (A =
      4) * (TD < 7) + 6 * (A = 4) *
      (TD > 6) + (A = 5) * (TD + 2
      ): PRINT A$: PRINT : PRINT :
       IF TD = 1 THEN BD = TD: IF
      A < 5 THEN  PRINT "BOTTOM NU
      MBER IS ALSO 1 DIGIT.": GOTO
```

眞正要動腦筋的小學四則運算

```
1420
1370  IF A < 5 THEN  PRINT "BOTTO
      M NUMBER 1 TO "BM" DIGITS >
      ";
1380  IF A = 5 THEN  PRINT "DIVID
      END "TD" TO "BM" DIGITS > ";

1390  POKE P,0: GET A$:BD =  PEEK
      ( - 16384) - 48: IF BD > BM OR
      BD < (A < 5) + TD * (A = 5) THEN
      1390
1400  PRINT A$
1410 R$ = "": IF A = 5 THEN  PRINT
      : PRINT : PRINT "DO YOU WANT
       REMAINDERS? (Y/N) > ";: POKE
      P,0: GET R$: PRINT R$:T = TD
      :TD = BD:BD = T
1420  PRINT : PRINT : PRINT "IS T
      HIS ALL CORRECT? (Y/N) > ";:
       POKE P,0: GET A$: PRINT A$:
       IF A$ <  > "Y" THEN 1230
1430  HOME : VTAB 10: POKE P,0: PRINT
       TAB( 3)"WANT TO SEE THE KEY
       LIST? (Y/N) > ";: GET A$: PRINT
      A$: IF A$ = "Y" THEN  HOME :
       GOSUB 1640
1440  HOME : IF A = AA OR A < 4 AND
      AA < 4 THEN  POKE  - 16299,0
      : POKE  - 16304,0:AA = A: GOTO
      1550
1450 AA = A: HGR2 : HCOLOR= 3: IF
      A < 4 THEN X = 14:Y = 6: HPLOT
      0,0 TO 279,0 TO 279,59 TO 0,
      59 TO 0,1 TO 278,1 TO 278,58
       TO 1,58 TO 1,1
1460  IF A < 4 THEN S$ = "PRESS
        R          TO REGROUP.": GOSUB
      70:Y = Y + 8:S$ = "PRESS
      D          TO MOVE DOWN": GOSUB
      70:Y = Y + 8:S$ = "PRESS
      " +  CHR$ (107) + "
      TO MOVE RIGHT.": GOSUB 70
1470  IF A < 4 THEN Y = Y + 8:S$ =
      "PRESS     " +  CHR$ (108) +
      "          TO MOVE LEFT.": GOSUB
      70:Y = Y + 8:S$ = "PRESS
      SPACE      TO ERASE.": GOSUB
      70:Y = Y + 8:S$ = "PRESS
      RETURN     WHEN YOU'RE DONE."
      : GOSUB 70
1480  IF A = 4 THEN X = 2:Y = 2: HPLOT
      0,0 TO 136,0 TO 136,75 TO 0,
      75 TO 0,1 TO 135,1 TO 135,74
       TO 1,74 TO 1,1:S$ = " R    =
        REGROUP": GOSUB 70:Y = Y +
      8:S$ = " " +  CHR$ (107) + "
        =   MOVE RIGHT": GOSUB 70
1490  IF A = 4 THEN Y = Y + 8:S$ =
      " " +  CHR$ (108) + "  =
      MOVE LEFT": GOSUB 70:Y = Y +
      8:S$ = " U    =   MOVE UP": GOSUB
      70:Y = Y + 8:S$ = " D    =
      MOVE DOWN": GOSUB 70
1500  IF A = 4 THEN Y = Y + 8:S$ =
      " L    =   DRAW OR": GOSUB 70
      :Y = Y + 8:S$ = "          ER
      ASE LINE": GOSUB 70:Y = Y +
      8:S$ = "SPACE  = ERASE": GOSUB
      70:Y = Y + 8:S$ = "RETURN =
      FINISH": GOSUB 70
1510  IF A = 5 THEN X = 182:Y = 5
      0: HPLOT 175,46 TO 279,46 TO
      279,191 TO 175,191 TO 175,47
       TO 278,47 TO 278,190 TO 176
      ,190 TO 176,47:S$ = "R  REGR
      OUP": GOSUB 70:Y = Y + 14:S$
       = "T  TEST OUT": GOSUB 70
1520  IF A = 5 THEN Y = Y + 8:S$ =
      "     AND": GOSUB 70:Y = Y +
      8:S$ = "   END TEST": GOSUB
      70:Y = Y + 14:S$ = "U  MOVE
      UP": GOSUB 70:Y = Y + 8
1530  IF A = 5 THEN S$ = "   ERAS
      E LINE": GOSUB 70:Y = Y + 14
      :S$ = "D  MOVE DOWN": GOSUB
      70:Y = Y + 8:S$ = "   DRAW L
      INE": GOSUB 70:Y = Y + 14:S$
       =  CHR$ (107) + "  MOVE RIG
      HT": GOSUB 70
1540  IF A = 5 THEN Y = Y + 14:S$
       =  CHR$ (108) + "  MOVE LEF
      T": GOSUB 70:Y = Y + 14:S$ =
      "SPACE - ERASE": GOSUB 70:Y =
      Y + 14:S$ = "RETURN - DONE":
       GOSUB 70
1550  IF A < 4 THEN AX = 126 + 14
       *  INT (TD / 2):YY = 172:TY
       = 88:BY = 104:AY(1) = 120:A
      X(1) = AX - 14 * (TD):AY(0) =
      144:E(1) = (266 - AX(1)) / 1
      4
1560  IF A = 4 THEN AX = 210 + 14
       *  INT (TD / 2):YY = 100:TY
       = 16:BY = 32: FOR T = 1 TO
      7:AY(T) = 40 + T * 16:AX(T) =
      AX - 14 * (TD + T - 1):E(T) =
      (266 - AX(T)) / 14: NEXT :AY
      (0) = 176:AX(7) = AX(6):E(7)
       = E(6)
1570  IF A = 5 THEN AX = 70 + 14 *
      TD:YY = 140:TY = 56:BY = 56:
      AY(0) = 0:AY(1) = 24:AY(2) =
      72:AY(3) = 96:AY(4) = 112:AY
      (5) = 136:AY(6) = 152:AY(7) =
```

眞正要動腦筋的小學四則運算

```
     176: FOR T = 1 TO 7:AX(T) =
     84:E(T) = (266 - AX(T)) / 14
     : NEXT
1580 SX = AX(1):LE = E(1): GOTO 5
     40
1590  TEXT : HOME : VTAB 5: INVERSE
     :S$ = " TOTALS ": GOSUB 60: HTAB
     S: PRINT S$: NORMAL : VTAB 8
     : HTAB 7 -  LEN ( STR$ (NR(1
     ))): PRINT NR(1);: HTAB 10: PRINT
     "RIGHT THE FIRST TRY."
1600  VTAB 10: HTAB 7 -  LEN ( STR$
     (NR(2))): PRINT NR(2);: HTAB
     10: PRINT "RIGHT THE SECOND
     TRY.": VTAB 15: PRINT "A TOT
     AL OF "NR(1) + NR(2)" RIGHT
     OUT OF "W - 1 - (K = 27)
1610  VTAB 18: PRINT "DO YOU WANT
      TO DO MORE? (Y/N) > ";: POKE
     P,0: GET A$: PRINT A$: IF A$
      <  > "Y" THEN  HOME : END
1620  VTAB 20: PRINT "THE SAME KI
     ND OF PROBLEMS? (Y/N) > ";: POKE
     P,0: GET A$: PRINT A$: HOME
     : IF A$ = "Y" THEN  POKE  -
     16299,0: POKE  - 16304,0: GOTO
     540
1630  GOTO 1230
1640  INVERSE : PRINT "KEY";: HTAB
     17: PRINT "FUNCTION": NORMAL
     : POKE 34,2: PRINT : PRINT "
     R     -MOVES THE CURSOR TO T
     HE RIGHT                PLACE
     FOR REGROUPING."
1650  IF A > 3 THEN  PRINT : PRINT
     "U     -MOVES THE CURSOR UP
     FROM ONE AN-            SWER
     LINE TO ANOTHER";: IF A = 5 THEN
      PRINT " ERASING
     UNUSED PROBLEM UNDERLINES";
1660  IF A > 3 THEN  PRINT "."
1670  PRINT : PRINT "D     -MOVES
      THE CURSOR DOWN FROM RE-
               GROUPING";: IF A >
     3 THEN  PRINT " OR TO THE NE
     XT LINE";: IF A = 5 THEN  PRINT
     "             DRAWING THE UND
     ERLINES";
1680  PRINT ".": PRINT : PRINT "A
     RROWS-MOVE THE CURSOR RIGHT
     OR LEFT               WITHOUT
      ERASING ANYTHING."
1690  GOSUB 1760: HOME
1700  PRINT "SPACE -ERASES THE NU
     MBER OVER THE CUR-
       SOR OR ANY REGROUPED NUMBE
     R             ABOVE IT. THE
     CURSOR MOVES                TO
      THE RIGHT."
1710  IF A = 4 THEN  PRINT : PRINT
     "L     -DRAWS OR ERASES THE
     LINE UNDER                THE P
     ROBLEM. IT IS DRAWN AT
            THE LEVEL OF THE CURS
     OR. IT                 CAN BE ER
     ASED FROM ANY SPOT."
1720  IF A = 5 THEN  PRINT : PRINT
     "T     -SETS UP AND RETURNS
     YOU FROM A              MULTI
     PLICATION PROBLEM UNDER
            THE DIVISOR. IT IS US
     ED TO                  TEST DIGI
     TS FOR THE ANSWER."
1730  PRINT : PRINT "RETURN-USE T
     HIS ONLY WHEN YOU ARE DONE.
          I WILL THEN SEE IF
      YOUR AN-              SWER I
     S CORRECT."
1740  PRINT : PRINT "ESC    -THIS
     STOPS THE PROBLEMS. USE IT
            IF YOU WANT TO QUI
     T OR CHANGE             THE KI
     ND OF PROBLEMS."
1750  PRINT : PRINT "?      -SHOWS
      YOU THIS LIST.": POKE 34,0
1760  VTAB 22: PRINT "** COPYRIGH
     T 1984 BY COMPUTING - AGE. *
     *": VTAB 23: PRINT  TAB( 9)"
     PRESS A KEY TO CONTINUE": POKE
     P,0: FOR T = 1 TO 2:TT =  RND
     (1):T = 1 + ( PEEK ( - 16384
     ) > 128): NEXT T: HOME : RETURN

1770  POKE P,0:LN =  PEEK (218) +
      PEEK (219) * 256:ER =  PEEK
     (222)
1780  IF ER = 254 OR ER = 255 THEN
      RESUME
1790  IF ER = 6 THEN  HOME : VTAB
     10: HTAB 8: FLASH : PRINT "
     SHAPE TABLE NOT ON DISK ": PRINT
     : HTAB 2: PRINT " INSERT COR
     RECT DISK AND PRESS A KEY";:
      NORMAL : GET A$: PRINT : HOME
     : GOTO 10
1800  IF ER = 8 THEN  HOME : VTAB
     10: HTAB 12: FLASH : PRINT "
      DRIVE DOOR OPEN! ": PRINT :
      HTAB 6: PRINT " PLEASE CLOS
     E AND PRESS A KEY!";: NORMAL
     : GET A$: PRINT : HOME : GOTO
     10
1810  TEXT : PRINT : PRINT "ERROR
      NUMBER "ER" IN LINE "LN".":
      PRINT : PRINT "PROGRAM IS E
     NDED.": END
```

眞正要動腦筋的小學四則運算

```
PRESS      R        TO REGROUP.
PRESS      D        TO MOVE DOWN
PRESS      →        TO MOVE RIGHT.
PRESS      ←        TO MOVE LEFT.
PRESS      SPACE    TO ERASE.
PRESS      RETURN   WHEN YOU'RE DONE.
```

```
BLOAD MATH CHARACTER
*6000.6525

6000- 52 00 A6 00 BA 00 C1 00
6008- DA 00 E8 00 00 01 27 01
6010- 2D 01 33 01 41 01 52 01
6018- 60 01 6E 01 72 01 79 01
6020- 80 01 8F 01 98 01 9E 01
6028- A4 01 A9 01 B2 01 BC 01
6030- C4 01 CF 01 DA 01 E3 01
6038- EF 01 FC 01 04 02 10 02
6040- 1D 02 22 02 28 02 31 02
6048- 39 02 42 02 49 02 5C 02
6050- 69 02 75 02 80 02 8B 02
6058- 97 02 A1 02 AD 02 B8 02
6060- C0 02 C9 02 D5 02 DC 02
6068- E8 02 F4 02 FE 02 08 03
6070- 14 03 21 03 2E 03 36 03
6078- 40 03 49 03 55 03 60 03
6080- 69 03 74 03 7C 03 84 03
6088- 8C 03 94 03 B9 03 DA 03
6090- F5 03 15 04 37 04 53 04
6098- 77 04 9C 04 B3 04 D8 04
60A0- FC 04 0A 05 18 05 92 92
60A8- 92 92 12 2D 2D 2D 2D 2D
60B0- 2D 35 3F 3F 3F 3F 3F 3F
60B8- 3F 00 92 92 2A 2D 2D 2D
60C0- 00 92 49 49 35 37 35 FF
60C8- 9B 2D 2D 2D 2D 35 3F 3F
60D0- 3F 3F 37 49 29 3E 2E 3E
60D8- 07 00 92 92 4A 2D 2D 2D
60E0- 2D 35 3F 3F 3F 3F 3F 00
60E8- 92 09 2E 2E 2E 2E 3E 3E
60F0- 3E 3E 2E 49 49 09 3C 3C
60F8- 3C 3C 2C 2C 2C 2C 2C 00
6100- 40 C0 40 C0 2D 15 3F 0E
6108- 2D 15 3F 0E 2D 15 3F 0E
6110- 2D 3E 2E 3E 2E 3E 2E 3E
6118- 2E 3E 2E 3E BF 2D 1E 3F
6120- 17 2D 1E 3F 17 2D 00 49
6128- 36 36 16 06 00 09 36 0D
6130- 24 04 00 09 36 77 3E 0E
6138- 6E 3C 28 E5 07 28 E5 24
6140- 00 49 3E 1E 0E 0C 05 A8
6148- DE 2E 15 1E 1C 36 07 E0
6150- 04 00 F1 0E 0C 4D 1E 1E
6158- 1E 17 4D 0C 15 1E 07 00
6160- 09 AD DF 6E 1E 37 0D 0D
6168- 1E DF 0E 6D 05 00 49 36
6170- 3B 00 49 F1 36 36 0E 06
6178- 00 09 0E 36 36 1E 06 00
6180- 11 0D 0D 1E 3F 17 2D 2D
6188- 1E 3F 17 0D 0D 05 00 49
6190- 12 FE 2A 2D F5 33 06 00
6198- 49 92 12 BE 06 00 92 0A
61A0- 2D 2D 05 00 49 92 92 06
61A8- 00 49 89 1E 1E 1E 1E 1E
61B0- 06 00 4A 2D 15 36 F6 3F
61B8- 1C 24 24 00 52 61 36 36
61C0- 17 2D 05 00 52 0C 2D 15
61C8- F6 BF 17 2D 2D 05 00 52
61D0- 0C 2D 15 9F AD F6 3F 1C
61D8- 07 00 0A 36 2D B5 0A 24
61E0- 24 24 00 49 89 3F 3F 36
61E8- 2D AD F6 3F 1C 07 00 92
61F0- 52 0E 2D 0C 1C 3F 27 64
61F8- 2D 15 06 00 0A 2D 2D F6
6200- 1E 1E 36 00 52 0C 2D 15
6208- 1E 3F 17 76 2D 0C 2C 00
6210- 92 52 0E 2D 0C 24 E4 3F
6218- 17 0E 2D 04 00 49 12 96
6220- 32 00 49 12 96 36 3B 00
6228- 49 F1 1E 1E 0E 0E 0E 06
6230- 00 92 29 2D B5 3F 3F 07
6238- 00 09 0E 0E 0E 1E 1E 1E
6240- 06 00 51 0C 15 BE B6 06
6248- 00 09 2D 15 DF BB 4D 6D
6250- 36 3E 1C BC 27 1F 36 0E
6258- 0E 2D 2D 00 92 31 36 4D
6260- 21 24 1C 1C 17 16 2D 04
6268- 00 52 2C 2D 15 17 3F 37
6270- 36 2D 65 24 00 4A 2D 15
6278- DF 33 36 0E 2D 0C 04 00
6280- 0A 36 36 2E 2D 0C 24 E4
6288- 3F 04 00 0A 2D 2D D6 3F
6290- 27 97 31 2E 2D 2D 00 0A
6298- 2D 2D D6 3F 27 97 31 36
62A0- 00 4A 2D 15 DE 2A 35 1E
62A8- 3F 1C 24 2C 00 0A 36 36
62B0- 6E 09 24 24 BC 3A 27 00
62B8- 4A 2D 1E 36 AE 3F 07 00
62C0- 0A 2D 2D DE 36 F6 1C 24
62C8- 00 0A 36 36 6E 09 1C 1C
62D0- 67 0C 0C 04 00 0A 36 36
62D8- 2E 2D 2D 00 0A 36 36 6E
62E0- 09 24 24 BC 1E 1C 04 00
62E8- 49 89 36 36 3E E0 E4 1C
62F0- 36 36 36 00 52 36 76 2D
62F8- 0C 24 E4 3F 07 00 92 21
6300- 2C 2D 15 F6 3F 37 36 00
6308- 52 36 76 6D 1C 0C 24 1C
6310- 3F 96 21 00 0A 36 36 6E
6318- 09 1C 3C 67 09 E4 3F 04
6320- 00 92 52 0E 2D 0C 1C E7
6328- 1C 0C 2D 15 06 00 0A 2D
6330- 2D DE 36 36 06 00 0A 36
6338- 36 0E 2D 0C 24 24 04 00
6340- 0A 36 76 0E 0C 0C 24 24
6348- 00 0A 36 36 0E 0C 0E 0C
6350- FC 4C 24 04 00 0A 0E 96
6358- 17 4D E1 1C 64 0C 04 00
6360- 0A 6E 48 F6 1F 0E 36 06
6368- 00 0A 2D 2D F6 1E 1E 17
6370- 2D 2D 05 00 09 2D DE 36
6378- 36 2E 2D 00 0A 0E 0E 0E
6380- 0E 0E 06 00 09 2D 36 36
6388- 36 3F 07 00 92 0A 0C 0C
6390- 0E 0E 06 00 52 29 2D 2D
6398- 2D 15 3F 3F 3F 3F 2E 3E
63A0- 2E 3E 36 36 36 25 24 24
63A8- 25 2D 2D 65 24 35 F6 1E
63B0- 3F 3F 0E 35 35 35 35 35
63B8- 00 52 49 2D 2D AD 3F 3F
63C0- 3F 37 37 35 37 35 37 35
63C8- 37 35 35 25 35 25 35 25
63D0- 25 25 27 25 27 25 27 25
63D8- 3F 00 52 49 09 35 3F 17
63E0- 2D 35 3F 3F 4E 29 3E 2E
63E8- 3E 2E 3E DF 2A 2D 2D 35
63F0- 3F 3F 3F 07 00 52 49 2D
63F8- 2D AD 3F 3F 3F BF 2E 6C
6400- 49 09 35 37 37 37 37 37
6408- 37 BF 2D 2D 2D 2D 3E 3F
6410- 3F 3F 3F 07 00 52 49 2D
6418- 2D AD 3F 3F 3F BF 2E 6C
6420- 49 09 35 37 F5 3F 3F 2E
6428- 2D 35 35 37 F5 37 27 37
6430- 27 37 27 27 27 2D 00 52
6438- 29 3E 2E 3E 2E 3E 2E 25
6440- 35 25 35 25 35 25 24 24
6448- 2C 36 36 36 36 36 3E 24
6450- 24 04 00 52 29 2D 2D 2D
6458- 2D 3E 3F 3F 3F 3F 2E 3E
6460- 2E 3E 2E 25 35 25 35 25
6468- 35 35 35 37 F5 37 27 37
6470- 27 37 27 27 27 2D 00 52
6478- 49 2D 2D 6D 12 27 3F 3F
6480- 3F 37 37 35 37 2D 2D 2D
6488- 35 3F 3F 3F 37 35 37 35
6490- 35 25 35 25 35 25 25 24
6498- AC 36 06 00 92 09 2C 2E
64A0- 2C 2E 2C 2E 2C 2E 2C 36
64A8- 37 F5 37 37 37 37 37 37
64B0- 35 37 00 52 49 2D 2D AD
64B8- 3F 3F 3F 37 37 35 77 2D
64C0- 2D 2D 3E 3F 3F 37 37 35
64C8- 77 35 25 35 25 35 25 25
64D0- 25 27 E5 24 25 27 25 00
64D8- 52 49 2D 2D AD 3F 3F 3F
64E0- 37 37 35 77 35 25 35 25
64E8- 35 25 25 24 35 36 3E 2E
64F0- 3E 2E 1E 37 27 37 27 37
64F8- 27 27 27 00 2E 2E 2E 2E
6500- 2E 2E 2E 2E 2E 2E 2E 2E
6508- 2E 00 92 29 2D 2D 2D 25
6510- 08 15 17 2D 1E 1E 06 00
6518- 4A B9 17 35 0E 04 28 2D
6520- 2D 2D 2D 05 00 FF
```

將INTEGER轉爲APPLESOFT

蘋果電腦最早期推出時是採用 INTEGER BASIC 爲語言的。但不久，即改用 APPLESOFT BASIC。兩種語言都各有特色。不過兩者却並不相容，需要作某些的更改才能運作。

現時我們接觸的程式已很少用 INTEGER BASIC 編寫。要執行 INTEGER BASIC 必須要使用 16 K RAMCARD。假如讀者未置有該咭則得到程式也不能用。爲此，本文旨在教導讀者如何將 INTEGER BASIC 轉爲 APPLESOFT 程式。

將 INTEGER程式轉變爲 APPLESOFT 程式第一個步驟是要做 TEXT FILE，過程非常簡單：（1）將 INTEGER 程式輸入記憶，（2）輸入以下四行：

```
32764  PRINT "dOPEN BASIC.TXT"
32765  PRINT "dWRITE BASIC.TXT"
32766  POKE 33,30: LIST 1,32763
32767  PRINT "dCLOSE BASIC.TXT": TEXT
     : END
```

"d" 代表 CTRL—D 以啓動所有 DOS 命令。事實上，在營幕上不會顯示出些甚麼，但輸入 CTRL—D 隨卽就出現第一個引號。

（3）按入 "GOTO 32764"，就會在程式中製造出 TEXT FILE 稱爲 BASIC·TXT，檔案可以重新給予名稱以便與原本 INTEGER程式的名稱更貼切。

現轉往APPLESOFT·EXEC TEXT FILE 入記憶系統，就會將原本的 INTEGER程式的句子翻譯成 APP—LESOFT 句子。但由於部份 INTEGER 句子無法爲 APPLESOFT 所明白，故此，程式還未可以運用。BASIC 的兩種"語言"有不同的語法結構，雖然如此，你仍可將那些句子輸入記憶，因爲在你正式 RUN 程式爲止之前，APPLESOFT 不會進行語法結構的審核。（INTEGER BASIC 則不同，輸入的句子，會第一時間被審核是否符合語法結構，所以將 APPLESOFT 程式轉爲 INTEGER 程式就困難得多。）

接着是 LIST 所有新的 APPLESOFT 程式，並更改任何因爲語法錯誤而不能運作的句子。雖然你可能很幸運，需要更改的事項很少，不過，以下列舉的十數種的更改，你可能隨時會遇到。

1. 長變數名稱

INTEGER 及 APPLESOFT 程式均容許長變數名稱（大多數的 BASIC 程式只准許兩個字符的變數名稱）。不過，由於 INTEGER BASIC 用整個長度去分辨不同的名稱，而 APPLESOFT 只用首兩個字符去分辨，因此，如果原本的 INTEGER 程式採用兩個易混淆的名稱，例舉如 PLAYER及 PLACE，你必須將其中一個名稱更改，以便 APPLESOFT 能正常運作。

2. KEY WORD的變數

含有 APPLE SOFT KEYWORD 的變數例如"DRAW"及"SINGLE"，都很可能在 INTEGER程式出現，但在 APPLESOFT 絕少用到（因爲 "SINGLE"首三個字母等如 APPLESOFT "SIN" 的功能）。

3. INTEGER 除算

小心程式中的除算運用，試看以下的例子：

```
100YX=XY／10+10*(XY—10*(XY／10))
```

這 INTEGER程式句子，XY，YX 掉轉運用，在譯爲 APPLESOFT程式，會易起混淆，應重新訂名爲XY%及YX%或用 INT 功能。

4. 字串的次元

在、INTEGER程式，每一字串一定要 DIMENSI—ONED至含有最高字符的數量（例如，"DIMA$(100)"卽表示字串A$可以有一百字符），在 APPLESOFT，你只須 DIMENSION 字串的矩陣（"DIM A $(100)"卽表示 100 字串A$（I），因爲I＝1至 100，每一個可有二百五十五個字符）所以，INTEGER程式轉爲 APPLESOFT 時，要移去所有DIM 字串。

5. 字串功能

在 INTEGER，如果你用 DIM 句子去聲明字串$，A$（I，J）所表達意思卽是 A$字串的字符位置由I到J。在APPLESOFT，同樣的意思是用 LEFT$（A$，J），因此，所有 INTEGER程式內如A$（I，J）的表達方式就要更改，並利用 APPLESOFT LEFT$，RIGHT$及MID$ 等功能。

兩個語言字串的結構及語言亦不相同。INTEGER程式句子可能有如下的例子：

```
100  A$(K)=B$
```

但轉爲 APPLESOFT 則要變成：

```
100  A$=LEFT$(A$,K—1)+B$
```

要注意你不能指定 LEFT$，MID$或 RIGHT$的價值，故此以下句子是不能執行。

```
102  RIGHT $(A$,7)="ORCHARD"
```

要改爲：

```
102  A$=LEFT $(A$,6)+"ORCHAR—D"。
```

6. IF 語句中的多行語句

假設，在 INTEGER程式有以下句子。

```
100  N=A:IF C>D THEN N=B:M=N+1
```

這句子在 APPLESOFT 運行可能有不同的結果，因爲當C大過D，M就不等如A＋1。原因是：當 INTEGE—R IF TEST失敗，就會執下一句子。該句子可能在同一行。但當 APPLESOFT IF TEST 失敗，就會執行下一行而非下一句子。

上述假設例子，應作以下更改以務求運作準確

```
100  N=A :IF C>D THEN N=B
101  M=N+1
```

7. IF 語句中的不等符

在 INTEGER BASIC，"A＃B"意思是"A不等如B"在 APPLESOFT ，應改爲"A＜＞B"。

8. RND 功能

INTEGER 及APPLESOFT 均有 RND 功能，但操作不同。在 INTEGER BASIC RND（X）表示0與X之間的隨機整數。在 APPLESOFT RND（X）則表示0與1之間的隨機絕對數字。改爲 APPLESOFT，應要用 INT（X*RND（1））

9. MOD 功能

APPLESOFT 沒有 MOD 功能，但可以用 DEFI—NED 作出同樣的功能。

舉例： INTEGER 句子

```
100  M=N MOD A
```

在 APPLESOFT，可用M等於N MODULO A；換言之，M是O與（A－1）之間的 INTEGER， 亦是N÷A的餘數。因此 12 MOD 3 等如0；13 MOD 3 等如1；14 MOD 3 是 2；15 MOD 3 等如0。

INTEGER N MOD A，在 APPLESOFT 應寫爲：

（N－INT（N／A）*A）

假如，程式需要運算很多 MODULO 16的數值，你可在 APPLESOFT 訂出一個功能。

10 DEF FN MOD 16（N）＝ N—16*INT（N／16）。

第一百句子就變爲：

100 M=FN MOD 16（N）

10. TAB 及 HTAB

APPLESOFT的 TAB 功能與 INTEGER 的 TAB 大不相同，所有 INTEGER 程式內的"TABX" 需用"HTABX"句子替代。

11. INPUT 語句中的逗號

試看以下 INTEGER程式例子：

```
100  INPUT "YOUR AGE?",
```

注意那逗號，是不會爲 APPLESOFT 程式所接納，一定要將它改爲分號（;）。

12. GOTO 或 GOSUB

在 INTEGER程式，你可在 GOTO 或 GOSUB之後接着表達句子或單字符變數，例如：

```
100  GOTO 100+N*10
```

在 APPLESOFT ，只容許 GOTO或 GOSUB 之後加上行數。

在上述例子，如N是1，程式就會 GOTO 行數 110。如果N相等2，就 GOTO 120 行數，如N＝3，就 GOTO 130。APPLESOFT 相同的表達方式是 ON …… GOTO 及 ON…… GOSUB ，假定N是1至3之間的變數。 APPLESOFT 程式應該如下：

```
100  ON N GOTO 110，120，130
```

13. ASC 功能

INTEGER 及 APPLESOFT 同樣有 ASC 功能，它可以產生 ASC Ⅱ 字符的數值。不過，兩種語言所產生的數值却不同，需要加入邏輯性的變更。假定，A$是字串，ASC（A$）在 INTEGER程式的數值比之 APPLE—SOFT ASC（A$）多出 128 。考慮以下的 INTE—GER 程式句子。

```
100 X=ASC（A$）：IFX＜176 OR X＞185
THEN PRINT "NON—NUMERIC!!"
```

進行A$的相等試驗，由 ASC Ⅱ 0至9，在 APPLE—SOFT ，試驗X由 48 至 57 。

14. 零頁中的機械語言

部份 INTEGER程式會 POKE 簡短機械語言常式至0頁位置由0至31。但在 APPLESOFT ，這種情況，並不出現，因爲 APPLESOFT 本身差不多全部用盡0頁（0頁是蘋果記憶首二百五十六個位置。由0至255）。唯一不被 APPLESOFT所佔用的0頁位置是6至9及25至31。

爲了使採用0頁常式的程式能在 APPLESOFT 如常運行。要進行以下步驟：（1）用 MONITOR "L" 命令 DISASSEMBLE常式。（2）決定將它放在那個位置（"第三頁"會是上佳位置）。（3）改變任何常式中的絕對參考資料。（4）修改所有放置機械語言常式於記憶的 POKE 句子。（5）修改引起爭議的 POKE 句子。（6）修改 CALL 句子。

15. LOMEN 及 HIMEM 的運用

一些 INTEGER 程式有 LOMEM 句子去移動 SY—MBOL TABLE（當程式運作時，變數，排列等都儲於這裡）。正常情況， INTEGER BASIC 將 SYMBOL

TABLE儲存於由 2048起首的位置。將 LOMEM 調至高於此，機械語言常式可儲於該處。APPLESOFT 的 LO—MEM 句子功能並不如此。APPLESOFT 程式通常儲存由 2049 起首的位置。SYMBOL TABLE則緊接在最後一個程式之後，而 LOMEM 自動較正至程式的尾部。即是說 LOMEM不可以較正至較低的數值。否則 SYM—BOL TABLE 就會覆蓋入程式之中。最佳解決的辦法是利用 ƐLOMEM:，當召喚至 APPLESOFT 程式中，可將程式在記憶系統位置推上。

HIMEM 在 INTEGER 及 APPLESOFT不同的用法，亦製造了難題。INTEGER程式是儲存於 HIM—EM向下的記憶位置，如何翻譯器執行 HIMEM 句子，是低於 HIMEM 數值的話，它會將程式移動向下低於新位置。APPLESOFT 就並不如此，不過，當 APPLESOFT 程式用了字串，而如果是 HIMEM 是過低的話，程式便告"消失"，因爲 APPLESOFT 將字串儲存於 HIMEM 之後。

16. 特別的零頁位置利用

一些 INTEGER程式應用下列的 0 頁位置，作爲解譯程式操作用途。

位　址	用　途
74及75	現行 LOMEM 數值
76及77	現行 HIMEM 數值
202及203	程式開始地址
204及205	SYMBOL—TABLE終止地址
208及209	最後相關 SYMBOL 地址

APPLESOFT程式進行同樣的功用的位址如下：

位　址
103及104
115及116
103及104
109及110
131及132

不過，如果你想用第一個表的位址去將 INTEGER程式變爲 APPLESOFT，就要小心從事，因爲直接用 APP—LESOFT "相等"的數值去替代，可能不會如你所願的。改變程式的邏輯有時並不容易，要視乎個別情況而定。在某些情況，INTEGER的數值很難爲 APPLESOFT 所替代，因爲每一種語言有其獨特的記憶結構。

17. LIST 語句中的變數

你可在以下的 INTEGER程式加入"非法"的句子。

```
100  LIST LINE 1，LINE 2
```

這句子在 APPLESOFT 不能運行，因爲 APPLE—SOFT 只接受 LIST 命令的行數，而非變數。

18. CLR 及 DEL 語句

這些"非法" INTEGER BASIC 句子亦可以加入程式中。APPLESOFT 命令相對於 "CLR" 是 "CL—EAR"。再者，INTEGER程式內的 DEL 句子在 APP LESOFT 運作不同。用於 APPLESOFT，在取消行數之後，程式執行會立即 STOP ，並會取消用不同語法的行數。例如， INTEGER句子的 " DEL 20" ，在 APPLE—SOFT應該變成 " DEL 20，20 "。

19. 大過32767的整數

INTEGER 程式正常情況是限制使用少於 32768 個數值，但一些程式一定需要使用超過 32767 個數值（例如48K機器的地址），在此情況下， INTEGER 程式必須用 PEEK 及 POKE 將數值分爲高低數元組。但由於 APP—LESOFT 用眞實算式去操作其功能，故此就無須有此顧慮，邏輯上較爲簡化。

20. SYMBOL TABLE 的 POKE 入

你可能遇到一些 INTEGER 程式會有以下形式去刺激 APPLESOFT CHR$ 功能，（1）決定一個字串成爲 SYMBOL TABLE 第一個變數，所以它的位置就很清楚（例如，如果 $ 是程式第一個決定的符號，而 LOMEM 並沒有修正，則 A$ 第一個字符的位置便是 2053），（2）POKE，ASC Ⅱ 數值由點數位轉爲已知的位置，（3）PRINT A$等等。要避免這種情況，就困難得多。

21. CALL 解譯程式

少數的程式， INTEGER BASIC INTERPRE—TER 內的 CALL 常式有不同的目的。例如在 INT—EGER 程式，RUN 的功能可以用 CALL—4116。LIST 功能可以 CALL—8117，在這些情況，是需要很高的知識去瞭解 INTERPRETER常式如何運作。以 " RUN " 的例子而言，在 APPLESOFT ，只需用 CALL 54630 替代，但替代 " LIST " 則仍無法找出答案。■

本刊第五期「濃縮圖片儲存空間」（P.70）是一個 INTEGER BASIC 程式，各讀者可以用此作爲嘗試實踐本文的更改方法，看看能否成功。我們歡迎讀者將轉爲 APPLESOFT 後的程式寄來本社，程式一經發表，即當致以薄酬。

澳洲製造商一部驕人產品

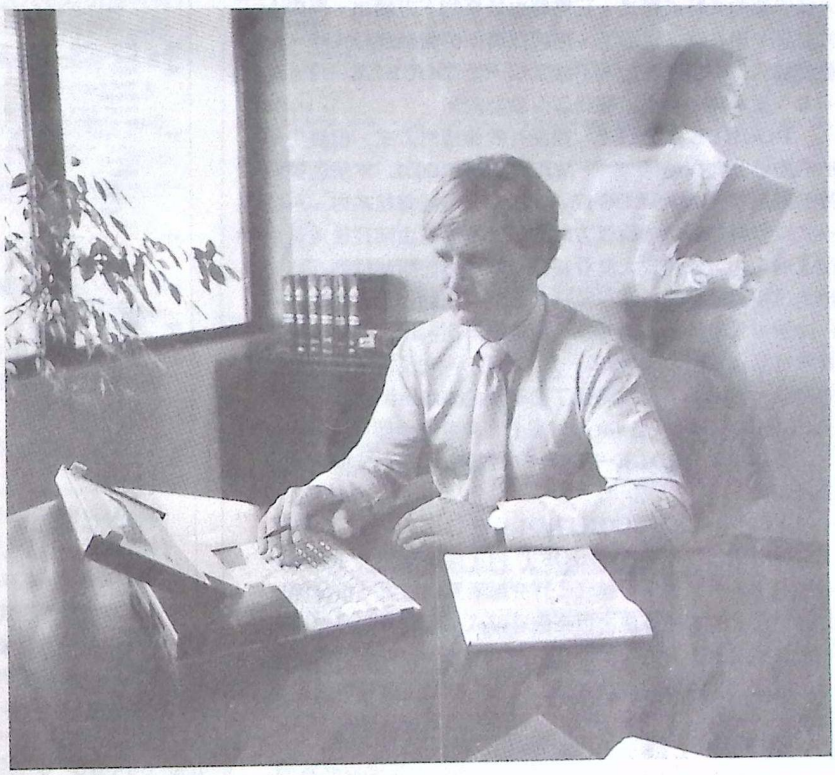
「多能」智多星手提電腦

在電腦之競爭市場中，電子製造商們希望能推出手提電腦而在其中佔一份兒，對於手提電腦之標榜形容各有不同。有些就商業生產規限而未能達到理想之設計，有些試圖以多種款式推出，但始終未能成功，或是製成外型輕便之電腦，但實際功能仍未能滿足一些需要多種用途之使用者。

一間於早期由政府科學及工業局資助之澳洲公司 DULMORT，爲眞正手提電腦找到答案。在最近推出一具手提而效能極高之電腦，其功能比一般案頭電腦更大。

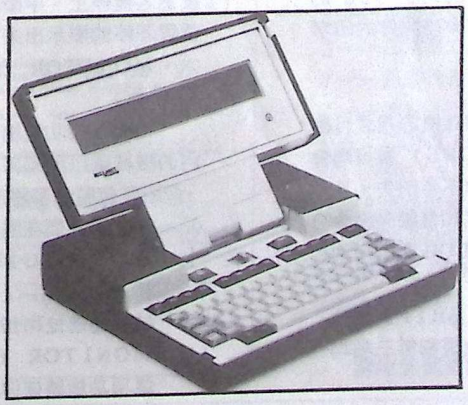
「多能」智多星採用了最新的 INTEL 80186 微型處理器，使各方面之表現比採用 INTEL 8088微型處理器的電腦大有改進，智多星麻雀雖小，但內存量可媲美任何 16 BIT 的商用小型電腦，它擁有 256 K CMOS RAM，128 K ROM 操作系統及其他一些應用軟件記憶。在這之上，還可加上兩個存量共 128 K BYTE 的記憶插件，以適合不同用戶之需要。

智多星採用時下非常流行的 MS DOS 操作系統，由 ROM直接操作，在設計上，CMOS RAM可當普通磁碟機，使用起來很方便。隨機內藏軟件包括有鬧鐘、日曆、計劃表、日記簿、文字處理器、計算表格，當然少不了流行之 MICRO－SOFT BASIC 語言，自然地，其他用 MS DOS語言的程式都可應用在智多星上。



鍵盤設計及大小均與普通之英文打字機相同，並附加一套十三個功用程序選擇之按鍵。

此具可在任何地方使用之電腦，顯像由液晶體所顯示，畫面可顯示八行，每行八十個字母，大小楷可由全備之按鍵控制，機身重量爲 3.5公斤。



智多星更可利用一些容易安裝之通訊組件，通過電話綫與其他地方之另一部電腦通訊。

一具如此細小的電腦預計將會成爲專業人士不可缺少之助手，試想一個記者要到外地採訪，當他獲得消息後，立刻可以直接將資料或文章如普通打字般存於電腦，利用文字處理程式，完成初稿或再稿以至到最後階段，他只需要利用電話機接駁到他的辦公室電腦中，直接將資料輸送回去。既可節省時間，並減少一些可能的錯誤。或試想一個經常到外地的營業人員，時常需要帶多種不同資料或一些商人需要收取一些保密之財經文件等。

一部「多能」智多星電腦售價爲港幣一萬八千九百五十元正。■

有廿四種功能的DOUBLE TAKE

DOUBLE－TAKE 主要是用來處理下列情況：每天當用電腦時，可能會使用 CATALOG或 LIST好幾百次，找你要的資料或程式，它們將由螢幕的下方顯出，再由上方消失，但在許多情形下，你想找前面的資料或程式時，便要再鍵入指令從頭開始，不勝其煩。而 DOUBLE－TAK－E 便是爲了解決這個問題而設計的。

DOUBLE－TAKE 裡面有數個應用方式，但最主要的功能是雙向移動（ 2 － WAY－SCROLL）。所謂雙向移動即是在 CATALOG或 LIST 或是監督程式內，可使程式或資料的顯示依兩種方向移動，不論是由前往後或是由後往前均可，像用放大鏡看報紙一樣，方向隨諸位的高興。控制這種功能只需用到←、→、ESC 三鍵即可。左鍵（←）控制程式向後，右鍵（→）則控制向前，ESC 則使程式暫停移動（當然， CTRL－S 亦可）。有了這雙方向移動的功能，我們再不担心錯過了要尋找的資料後要再來 CAT－ALOG或 LIST 一次，只需令它向前捲動即可。

此外， DOUBLE－TAKE 還提供了二十三種有用的功能：

一、NEW LIST（CONTROL－F1）

我們知道在 APPLESOFT 內，同一行號（ LINE NUMBER）下可用冒號（：）分隔不同的指述。這樣固然對設計程式的人很方便，但在觀看程式的時候可就沒那麼輕鬆，整個顯示螢幕塞滿了指令、指述，若無印字機而只從螢幕上看就眞令人眼花。 NEW LIST的功能便是將程式簡化，一行一段敍述（見表一），如此看起來便清晰得多了。

二、APPLE LIST（CONTROL－F!）

在正常情況下 LIST用 CTRL－F 和!見表二（應注意直接鍵入 LIST 將無法使用雙向移動的功能）。此外，若配合印字機列印，上述兩種 LIST 的功能將更強，我們只需 PR＃SLOT ，再用下列的指述： LIST〔LI－NE＃1〕－〔 LINE＃2 〕〔／ PRINT WIDTH〕便可印出50、60、70等寬度的報表。表三和表四就是用 LI－ST／80 所印出的80字行報表。當然，只要印字機夠大，你要印多寬就可以印多寬，它可讓你印出／20到／ 255 的寬度，這樣就毋需再用 POKE 一大堆數值來控制你的印字機了。

三、CATALOG D1（CONTROL－F2）

使用 CTRL－ F和2 可觀看第一部磁碟機的程式目錄，同時磁碟容量和未使用的磁區（ SECTORS）數目均會顯現在螢幕的第一行，即使目錄向上捲動也不會消失，且被刪除的檔名會在原處留下空白，當新的程式加進磁碟後會自動補進第一個空白處，如直接鍵入 CATALOG亦能得到相同的效果，並可用雙向移動的功能。

四、CATALOG D2 (CONTROL－F SHIFT 2）

觀看第二部磁碟機的磁碟目錄，其餘效果和第三項均一樣。

```
]FUNCTION? LIST

10    REM
        HELLO (DOUBLE-TAKE)

20    PRINT CHR$ (12); CHR$ (21)
    : REM TURN OFF 80 COLUMN CARD
30    TEXT
    : HOME
    : NORMAL
    : PRINT CHR$ (4)"NOMONICO"
    : G$ = CHR$ (34)
40    IF PEEK (978) = 157 AND PEEK (401
        93) = 156 AND PEEK (1005) = 234
        THEN DN$ = ""
    : GOTO 90
50    IF PEEK (978) = 157 AND PEEK (401
        93) = 140 AND PEEK (1005) = 1
        THEN DN$ = ".48"
    : GOTO 90
60    IF PEEK (978) = 157 AND PEEK (401
        93) = 156 AND PEEK (1005) = 2
        THEN DN$ = ".LC"
    : GOTO 90
70    IF PEEK (978) ) = 190 THEN DN$
        = ".DM"
    : GOTO 90
80    PRINT
    : HTAB 12
    : PRINT "NON STANDARD SETUP."
    : HTAB 4
    : PRINT "BOOT THIS DISK TO LOAD DOU
      BLE-TAKE."
    : STOP
90    HTAB 10
    : PRINT "D O U B L E - T A K E"
    : PRINT
    : HTAB 15
    : PRINT "VERSION 1.0"
    : HTAB 8
    : PRINT " (C) 1983  MARK SIMONSEN"
    : PRINT
    : HTAB 15
    : PRINT "BEAGLE BROS"
    : HTAB 12
    : PRINT "4315 SIERRA VISTA"
    : HTAB 11
    : PRINT "SAN DIEGO, CA 92103"
100   VTAB 12
    : PRINT "  IMPORTANT:"
    : PRINT "  ----------"
    : PRINT
    : PRINT "* RUN THE "G$"NOTES"G$" PR
      OGRAM FOR THE"
    : PRINT "  LATEST UPDATES AND CHANG
      ES."
    : PRINT
    : PRINT "* RUN THE "G$"DOUBLE-TAKE
      DEMO"G$" PROGRAM"
    : PRINT "  FOR A DEMONSTRATION."
    : PRINT
110   PRINT "LOADING DOUBLE-TAKE"DN$"..
      ."
120   PRINT CHR$ (4)"BRUN DOUBLE-TAKE";
      DN$
    : END
```

表（1）

```
]FUNCTION? LIST

10  REM
      HELLO (DOUBLE-TAKE)

20  PRINT  CHR$ (12); CHR$ (21): REM
      TURN OFF 80 COLUMN CARD
30  TEXT : HOME : NORMAL : PRINT
      CHR$ (4)"NOMONICO":G$ =  CHR$
      (34)
40  IF  PEEK (978) = 157 AND  PEEK
      (40193) = 156 AND  PEEK (100
      5) = 234 THEN DN$ = "": GOTO
      90
50  IF  PEEK (978) = 157 AND  PEEK
      (40193) = 140 AND  PEEK (100
      5) = 1 THEN DN$ = ".48": GOTO
      90
60  IF  PEEK (978) = 157 AND  PEEK
      (40193) = 156 AND  PEEK (100
      5) = 2 THEN DN$ = ".LC": GOTO
      90
70  IF  PEEK (978) )  = 190 THEN
      DN$ = ".DM": GOTO 90
80  PRINT : HTAB 12: PRINT "NON S
      TANDARD SETUP.": HTAB 4: PRINT
      "BOOT THIS DISK TO LOAD DOUB
      LE-TAKE.": STOP
90  HTAB 10: PRINT "D O U B L E -
      T A K E": PRINT : HTAB 15: PRINT
      "VERSION 1.0": HTAB 8: PRINT
      " (C) 1983  MARK SIMONSEN": PRINT
      : HTAB 15: PRINT "BEAGLE BRO
      S": HTAB 12: PRINT "4315 SIE
      RRA VISTA": HTAB 11: PRINT "
      SAN DIEGO, CA 92103"
100 VTAB 12: PRINT "  IMPORTANT:
      ": PRINT "  ----------": PRINT
      : PRINT "* RUN THE "G$"NOTES
      "G$" PROGRAM FOR THE": PRINT
      "  LATEST UPDATES AND CHANGE
      S.": PRINT : PRINT "* RUN TH
      E "G$"DOUBLE-TAKE DEMO"G$" P
      ROGRAM": PRINT "  FOR A DEMO
      NSTRATION.": PRINT
110 PRINT "LOADING DOUBLE-TAKE"D
      N$"..."
120 PRINT  CHR$ (4)"BRUN DOUBLE-
      TAKE";DN$: END
```

表（2）

五、HEX／ASE Ⅱ DUMP（CONTROL F 3）

這項功能可使整個記憶區所放置的16進位機械碼和相等的 ASC Ⅱ符號同時顯示出來，除非你指定一個範圍，或是按下各種停止、中斷的鍵，否則整個記憶區將從指定的起始位置不停的顯示出來。

六、MONITOR DISASSEMBLE（CONTROL－F ＃）

和前述的功能相似， DISASSEMBLE 將指定範圍內的機械碼以滙編語言的形式出現，除非強迫它暫停或中斷，否則整個顯示幕將不停移動(不論是否用到雙向移動的功能），省却了我們在監督程式內不斷鍵入L的困擾。

七、HEX／DECIMAL CONVERT（CONTROL－F 4）

轉換16進位和10進位數值。

八、MONITOR BASIC（CONTROL F $）

這項功能將使你進入監督程式，輸入一行機械語言(最多

255個字），當按下 RETURN 鍵後將自動回到 BASIC 語言內。

九、APPEND HIDE（CONTROL－F 5）

十、APPEND MERGE（CONTROL－F %）

DOUBLE－TAKE 所提供的這兩項功能必須聯合使用才行。APPEND HIDE將原先存在記憶體內的程式隱藏起來，當你鍵入 CTRL－F 5 後，原先的程式會 LIST 不出來，隨後再鍵入或輸入第二個程式，按下 CTRL－F %，兩個程式便會依行號順序混合起來，此時若有錯誤發生，例如行號有重複等，螢幕上便會出現錯誤的訊息，並將兩個程式維持在原來的狀態，讓我們再作參考。讀者可以表五的例子做參考，便能瞭解。

十一、RENUMBER（CONTROL－F 6）

這項功能可使你的程式行號依自己的喜好重新排列。當你按下 CTRL－F 6 後，隨著螢幕的指示依次鍵入①原程式起始行號；②原程式結束的行號；③新的起始行號；④新行號的間隔後，DOUBLE－TAKE 便會將整個行號重新排列好，同時亦保存了原程式中 GOTO、GOSUB 等指令轉移行號的位置。表六便是重編號後所產生的結果。

十二、AUTO－NUMBER（CONTROL－F SHIFT 6

若你時常撰寫 BASIC 程式，這個功能將提供便利的自動編號。你只需按下 CTRL－F SHIFT 6，告訴起始行號和號碼的間隔，自動行號便可用 SPACE BAR 來進行。如此便不再需要自行輸入行號，只稍敲一下 SPACE BAR，DOUBLE－TAKE便會替你把這些工作完成。

十三、VITAL STATISTICS（CONTROL－F 7）

這項功能將系統所使用的記憶位址顯示出來，諸如程式的起始位址、結束位址等。（參看表七）

十四、2－BYTE PEEK（CONTROL－F SHIFT 7）

這項功能顯示出某一特定位址兩個 BYTE 的值，就等於我們輸入 PEEK（A）＋PEEK（A＋1）＊256 一樣。其中A代表位址。

十五、VARIABLE CROSS－REFERENCE（CONTROL－F 8）

表（4）

```
]FUNCTION? LIST /80

10  REM
    HELLO (DOUBLE-TAKE)

20  PRINT  CHR$ (12); CHR$ (21): REM TURN OFF 80 COLUMN CARD
30  TEXT : HOME : NORMAL : PRINT  CHR$ (4)"NOMONICO":@$ =  CHR$ (34)
40  IF  PEEK (978) = 157 AND  PEEK (40193) = 156 AND  PEEK (1005) = 234 THEN DN$
    = "": GOTO 90
50  IF  PEEK (978) = 157 AND  PEEK (40193) = 140 AND  PEEK (1005) = 1 THEN DN$ =
    ".48": GOTO 90
60  IF  PEEK (978) = 157 AND  PEEK (40193) = 156 AND  PEEK (1005) = 2 THEN DN$ =
    ".LC": GOTO 90
70  IF  PEEK (978) >  = 190 THEN DN$ = ".DM": GOTO 90
80  PRINT : HTAB 12: PRINT "NON STANDARD SETUP.": HTAB 4: PRINT "BOOT THIS DISK
    TO LOAD DOUBLE-TAKE.": STOP
90  HTAB 10: PRINT "D O U B L E - T A K E": PRINT : HTAB 15: PRINT "VERSION 1.0"
    : HTAB 8: PRINT " (C) 1983  MARK SIMONSEN": PRINT : HTAB 15: PRINT "BEAGLE
    BROS": HTAB 12: PRINT "4315 SIERRA VISTA": HTAB 11: PRINT "SAN DIEGO, CA 92
    103"
100 VTAB 12: PRINT "  IMPORTANT:": PRINT "  ----------": PRINT : PRINT "* RUN T
    HE "@$"NOTES"@$" PROGRAM FOR THE": PRINT "  LATEST UPDATES AND CHANGES.": P
RINT
    : PRINT "* RUN THE "@$"DOUBLE-TAKE DEMO"@$" PROGRAM": PRINT "  FOR A DEMONS
    TRATION.": PRINT
110 PRINT "LOADING DOUBLE-TAKE"DN$"..."
120 PRINT  CHR$ (4)"BRUN DOUBLE-TAKE"|DN$: END
```

表（3）

```
]FUNCTION? LIST /80

10    REM
      HELLO (DOUBLE-TAKE)

20    PRINT CHR$ (12); CHR$ (21)
    : REM TURN OFF 80 COLUMN CARD
30    TEXT
    : HOME
    : NORMAL
    : PRINT CHR$ (4)"NOMONICO"
    : @$ = CHR$ (34)
40    IF PEEK (978) = 157 AND PEEK (40193) = 156 AND PEEK (1005) = 234 THEN DN$
=
        ""
    : GOTO 90
50    IF PEEK (978) = 157 AND PEEK (40193) = 140 AND PEEK (1005) = 1 THEN DN$ =
      ".48"
    : GOTO 90
60    IF PEEK (978) = 157 AND PEEK (40193) = 156 AND PEEK (1005) = 2 THEN DN$ =
      ".LC"
    : GOTO 90
70    IF PEEK (978) > = 190 THEN DN$ = ".DM"
    : GOTO 90
80    PRINT
    : HTAB 12
    : PRINT "NON STANDARD SETUP."
    : HTAB 4
    : PRINT "BOOT THIS DISK TO LOAD DOUBLE-TAKE."
    : STOP
90    HTAB 10
    : PRINT "D O U B L E - T A K E"
    : PRINT
    : HTAB 15
    : PRINT "VERSION 1.0"
    : HTAB 8
    : PRINT " (C) 1983  MARK SIMONSEN"
    : PRINT
    : HTAB 15
    : PRINT "BEAGLE BROS"
    : HTAB 12
    : PRINT "4315 SIERRA VISTA"
    : HTAB 11
    : PRINT "SAN DIEGO, CA 92103"
100   VTAB 12
    : PRINT "  IMPORTANT:"
    : PRINT "  ----------"
    : PRINT
    : PRINT "* RUN THE "@$"NOTES"@$" PROGRAM FOR THE"
    : PRINT "  LATEST UPDATES AND CHANGES."
    : PRINT
    : PRINT "* RUN THE "@$"DOUBLE-TAKE DEMO"@$" PROGRAM"
    : PRINT "  FOR A DEMONSTRATION."
    : PRINT
110   PRINT "LOADING DOUBLE-TAKE"DN$"..."
120   PRINT CHR$ (4)"BRUN DOUBLE-TAKE"|DN$
    : END
```

記憶體內程式所用到的變數，依字母順序顯示出來，並將每個變數所出現的行號顯示出來。

十六、VARIABLE DISPLAY（CONTROL－F SHIFT 8）

顯示出的每個變數在記憶體內目前的值。

十七、REPLACE CURSOR（CONTROL－F 9）

利用這種功能，我們可選擇四種游標，按下 CTRL－F 9後，顯示幕會出現 CURSOR這個字，接着再鍵入各種游標所代表的號碼：

1——正常狀態下的游標。

2——正常狀態的形狀，但不再閃爍。

3——化方塊成一直綫（底綫），會閃爍。

4——直綫，但不會閃爍。

十八、CONTROL－CHARACTERS（CONTROL－F SHIFT 9）

這項功能將使你在螢幕上看到 CONTROL 字元，無論在 CATALOG 或是 LIST 等情況均可。若想消除這項功能只需再按 CTRL－F SHIFT 9 即可。

十九、CLEAR SCREEN（CONTROL－F 0（ZERO））

清除整個畫面，像 HOME 這個指令一樣。

有廿四種功能的 DOUBLE TAKE

二十、HI－RES PAGE 1（CONTROL－F：）
觀看高解像度第一頁的圖形。

廿一、LO－RES PAGE 1（CONTROL－F＊）
觀看低解像度第一頁的圖形。

廿二、PAGE 2（CONTROL－F－）
觀看第二頁內的資料。

廿三、SPLIT－SCREEN（CONTROL－F＝）
使用這項功能將使到原有的圖形顯示空出四行的位置，讓我們使用文字。

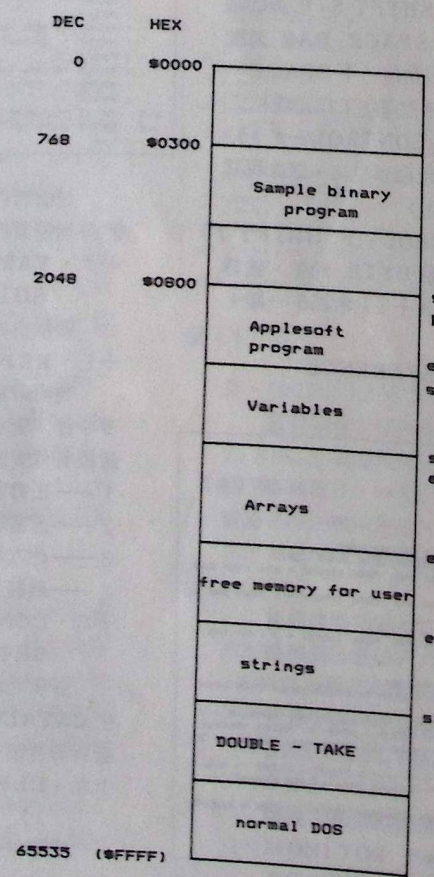
總括來說，這軟件在應用效果上確令用家感到非常方便和有效率。■

```
]NEW

]PR#0
]FUNCTION? LIST

10    REM   ? GROCERY LIST
20    TEXT
    : HOME
    : NORMAL
    : GOSUB 2000
    : END
30    DATA  DOGFOOD,PRETZELS,SODA POP
40    DATA  TV GUIDE,NATIONAL ENQUIRER
2000  FOR I = 1 TO 5
    : READ ITEM$
    : PRINT ITEM$
    : NEXT
    : RETURN

]PR#0
]FUNCTION? LIST

1000  REM  CENTER A$
1010  HT = INT (20 - ( LEN (A$) / 2))
        + 1
    : IF HT < 0 THEN HT = 1
1020  HTAB HT
    : PRINT A$
    : RETURN

]

]FUNCTION?
PROGRAMS MERGING...

]

]FUNCTION? LIST

10    REM   ? GROCERY LIST
20    TEXT
    : HOME
    : NORMAL
    : GOSUB 2000
    : END
30    DATA  DOGFOOD,PRETZELS,SODA POP
40    DATA  TV GUIDE,NATIONAL ENQUIRER
1000  REM  CENTER A$
1010  HT = INT (20 - ( LEN (A$) / 2))
        + 1
    : IF HT < 0 THEN HT = 1
1020  HTAB HT
    : PRINT A$
    : RETURN
2000  FOR I = 1 TO 5
    : READ ITEM$
    : PRINT ITEM$
    : NEXT
    : RETURN
```

表（5）

```
]PR#0
]FUNCTION? LIST

1     HGR
    : HCOLOR= 3
2     FOR I = 80 TO 160
    : HPLOT 0,I TO 279,I
    : NEXT
3     HCOLOR= 0
    : CX = 140
    : CY = 50
4     FOR I = 1 TO 11
    : READ X,Y
    : HPLOT CX,CY TO X,Y
    : NEXT
5     DATA  0,90,0.100,0,120,0,160,70,1
        60,140,160,210,160,279,160,279,
        120,279,100,279,90
6     I = 1
7     IF 80 + I ^ 2.5 > 160 THEN 10
8     HPLOT 0,80 + I ^ 2.5 TO 279,80
        + I ^ 2.5
9     I = I + 1
    : GOTO 7
10    END

]

]FUNCTION?

FIRST LINE TO BE RENUMBERED
(RETURN = START OF PROGRAM)
?

LAST LINE TO BE RENUMBERED
(RETURN = END OF PROGRAM)
?

STARTING LINE NUMBER
(RETURN = 10)
?

INCREMENT BETWEEN LINES
(RETURN = 10)
?20

1 --> 10
2 --> 30
3 --> 50
4 --> 70
5 --> 90
6 --> 110
7 --> 130
8 --> 150
9 --> 170
10 --> 190
WAIT...

]FUNCTION? LIST

10    HGR
    : HCOLOR= 3
30    FOR I = 80 TO 160
    : HPLOT 0,I TO 279,I
    : NEXT
50    HCOLOR= 0
    : CX = 140
    : CY = 50
70    FOR I = 1 TO 11
    : READ X,Y
    : HPLOT CX,CY TO X,Y
    : NEXT
90    DATA  0,90,0.100,0,120,0,160,70,1
        60,140,160,210,160,279,160,279,
        120,279,100,279,90
110   I = 1
130   IF 80 + I ^ 2.5 > 160 THEN 190
150   HPLOT 0,80 + I ^ 2.5 TO 279,80
        + I ^ 2.5
170   I = I + 1
    : GOTO 130
190   END
```

表（6）



表（7）

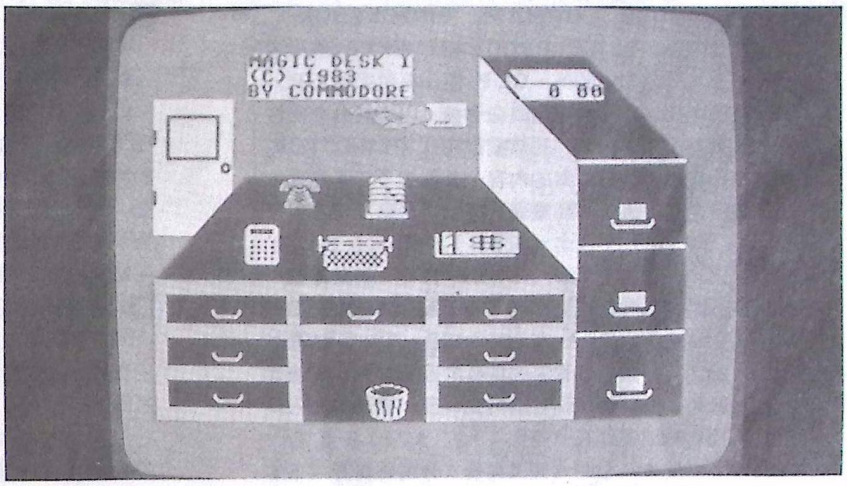
60元一個 COMMODORE MOUSE??

神奇寫字枱 MAGIC DESK I

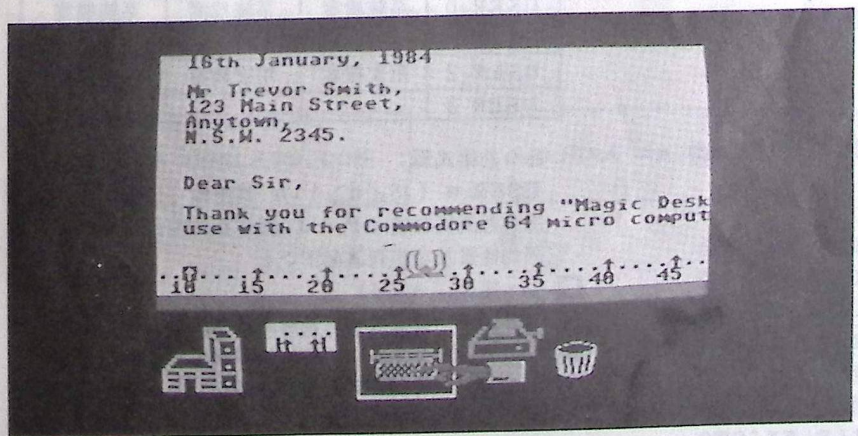
隨著電腦科技的發展，人類操作電腦的方式已由傳統的按鍵輸入發展到今天的MOUSE及TOUCH SCREEN等輸入方式。不過，這些先進的輸入設備一般都是所費不菲，令人望而却步。

美國COMMODORE 廠最近推出一個配合其C－64型電腦使用的文書及資料處理軟件——神奇寫字枱（MAGIC DESK），這個軟件的操作方式和時下的MOUSE 操作方式差不多，所不同者就是這個軟件不需配用昂貴的MOUSE設備，只需配用一個價值六十元的遊戲棒就可操作。

一般文字處理器的說明書都是洋洋千言，令到用者好不厭煩，但這個神奇寫字枱由於應用了MOUSE 的概念，把那些繁複的指令變作圖案，憑著圖案所顯示的意思，用者就可知道指令的意思。舉例來說，當你需要輸入文字內容時，只需將螢幕上顯示的一隻手，利用遊戲棒移往一部螢幕上顯示的打字機處。螢幕就會顯示出一部打字機的近鏡，而一張空白的紙亦會同時出現在打字機上面。這時，文字的輸入和操作就與一般的電動打字機沒有兩樣。當文稿做好經過核對後，只需用遊戲棒把「手」移往螢幕上的印字機處，文稿就會在一部連接好的印字機裏編印出來。在資料儲存方面，神奇寫字枱的功能更令用者省回很多檢閱資料的時間。假設一篇打好的文稿需要保存的話，只需把「手」移往螢幕上的一個文件櫃內的十個文件袋。用者這時可自行把要儲存的文件編名，放進這些文件袋裏便可。由於神奇寫字枱顯示檔案的方法和一般的辦公室文件櫃排列方法一樣，故翻查檔案特別容易。整個文件櫃可存三百張紙的資料。



（1） MAGIC DESK 1 的操作模 式選擇



（2）在輸入文字時，使用方法一如普 通的打字機

總結整個神奇寫字枱的操作過程而言，可以說是目前市面上可以購買到最實際及易用的文書及資料軟件。就算是一個未接觸過電腦的人士，也可以在十五分鐘內完全明白所有操作。

本軟件名爲 MACIC DESK 1
適合 COMMODORE 電腦使用

根據外電報道，COMMODORE 廠在今年還準備推出一個配合其 COMMODORE 64 型電腦使用的 TOUCH SCREEN顯示器。相信到時在港推出時，香港的微電腦市場又會有一番新景象。□

CP/M 4 入門基礎篇

隨機命令及工具程式

黃偉明

APPLE Ⅱ 的 CP／M操作系統之組成，除了隨機的六個命令外，還包括了十個工具程式。在第一期中我們提到的 CCP（CONSOLE COMMAND PROCESSOR）及 TPA（TRANSIENT PROGRAM AREA)就與它們有很密切且不可分割的關係。CCP 担任著用家與 CP／M操作系統的橋樑，它讀進了隨機命令後，因不同命令的性質，而作出如列出 CP／M 的檔案目錄（DIRECTOR—Y），在印表機上列出檔案的內容，及負起了工具程式在 TPA 的控制及調度工作，隨機命令包括了清除（ERA），檔案目錄（DIR），改名（REN），儲存（SAVE），列出（TYPE）及用家（USER）等。另外組成 CP／M操作系統的就是工具程式（TRANSIENT C—OMMAND），它可與TPA 有直接的關係，TPA 儲存了由磁碟讀入的工具程式，如程式在編輯的過程中，它就載著了程式的機械語言及資料，而當編輯完成後，經過了載入及執行後，一個新的工具程式便誕生了。在 APPLE Ⅱ 的 CP／M操作系統中，它的八個工具程式分別爲統計（ST—AT)，組合（ASM），載入（LOAD），除錯封裝（DDT），PIP，編輯（ED），SUBMIT，及DUMP 等。

■ 1、清除（ERA）

顧名思義，這個命令就是用作清除 CP／M 系統磁碟的檔案，正如 DOS 的 DELETE 一樣，它的格式爲；

ERA FILE NAME （名稱），TYPE(類別)

例子 1.1爲清除一個叫 USELESS.COM的檔案，例子 1.2 爲清除所有以A字爲首，類別屬於 COM 的檔案，在例子 1.2的"*"爲百搭符號。例子 1.3爲清除磁碟上所有的檔案，CP／M作出的反應爲叫用家核對是或否，（Y／N）？如答"Y"則所有檔案被清除。

■ 2、目錄（DIR）

DIR 使磁碟上所有合符檔案的名稱及類別的檔案列出，其格式爲：

DIR

例子 2.1爲最簡單的格式，例子 2.2 爲列出所有以A字爲首而類別爲 COM 的檔案，例子 2.3爲列出B：磁碟機的所有以D字爲首的檔案。

■ 3、改名（REN）

它使用家可以把磁碟上的檔案名稱及類別改做第二個名字，格式爲REN NEW.COM=OLD.COM。例子 3.1把一個叫 PIP.COM 的檔案改叫做 XYZ.COM，例子 3.2 把一個叫 STAT.COM 的檔案改叫做 STATT.AAA 。但請注意，在"相等符號"的左邊爲新的名字，在右邊爲舊的名字，而"相等符號"亦可由"←"所替代。假如要改的新名字已存在，CP／M 就會作出"FILE EXISTS"的反應，如例 3.3。假如要改的舊名稱，根本不存在，CP／M就會作出"NOT FOUND"的反應，如例 3.4。

■ 4、列出（TYPE）

TYPE 把 CP／M磁碟上所說明的檔案內容，（以 ASCⅡ 形式）列出，其途徑有二，第一個爲在螢幕上列出，第二個爲在列表機印出，要達致這個效果，用家要先行按＜CTRL—P＞才打入 TYPE 這個命令，要消除列表機的操作，可再按＜CTRL—P＞。它的格式爲：

TYPE FILENAME.FILETYPE

例 4.1爲在印表機上列出一個名爲 SCRAMBLE.AS—M的檔案。

■ 5、儲存（SAVE）

SAVE 是把儲存在 TPA 內的程式，以 256 字節爲一頁的形式儲存在磁碟上，而起址位址爲 100 H。它的格式爲：

SAVE n FILENAME.FILETYPE

例子 5.1爲把一個在地址 100 H 起至 2FFH 止的程式用 QQ.WW 的名稱儲存在磁碟上，例 5.2 爲把一個由地址 100H 開始至 28FFH止的程式，以SS.WW 的名稱儲存在B：磁碟機上。

■ 6、用家（USER）

USER 爲一個較少用到的命令，它使用家可以自由選擇在那一個用家邏輯區工作，當 CP／M 操作系統第一次 BOOT 起時，用家的邏輯區爲 0。用家可因不同的邏輯區而編輯不同種類的檔案，但當用家要把檔案儲存時，他必需把這個檔案用 PIP 的工具程式搬回零號邏輯區才行，而U—SER 這個命令，主要是爲方便有兩個或以上的磁碟機的用家而設，其組合可參看圖 6.1

	A:磁碟機	B:磁碟機	C:磁碟機
USER 0	系統檔案	系統檔案	系統檔案
USER 1			
USER 2	用家檔案	用家檔案	用家檔案
USER 3			

命令的格式爲：

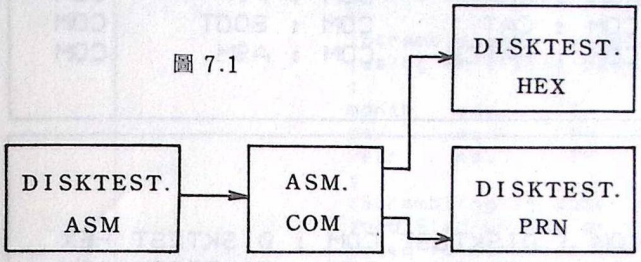
USER n （15≧n≧1，n 爲整數）

工具程式爲組成 CP／M 操作系統的另一個重要環節，它們的性質及功能列述如下：

■ 1、組合（ASM）

ASM這一個工具程式是用來驅動廸吉多電腦公司（DIGITAL RESEARCH）的滙編器，檔案的全名爲 ASM.COM。這個程式把 8080 或 Z 80 微處理器滙編語言寫成的原程序，加以翻譯而變爲機器語言，其過程如圖 7.1 所示。

圖 7.1



但請注意，原程式的類別必須是 ASM，如 DISK TEST.ASM 。它經過 ASM.COM 的翻譯後，原程式便化做兩個檔案，分別爲用機械語言組成的 DISKTEST. HEX 檔案及可在印字機上印出的 DISKTEST.PRN 檔案。在 DISKTEST·PRN 檔案中，它有原程式的字行號數，翻譯錯誤提示及機械簡碼等。而 DISKTEST.HEX 檔案可用載入（ LOAD ）這個工具程式來產生一個新的工具程式，LOAD 的操作方法會在後面論述。要執行 ASM. COM ，我們可以用以下的任何一格式；

（A）ASM FILENAME

（B）ASM FILENAME.PARMS

在這兩種程式中，用家只需給予檔案名稱便可，因爲它的類別已一早被設定爲 ASM。（A）的格式最簡單，正如圖 7.1的描述一樣，ASM 被呼叫去翻譯一個叫 DISKTE ST.ASM 的檔案，而產生的 DISKTEST.HEX 及 DI SKTEST.PRN 檔案均儲存在同一個磁碟機中。（B）的格式比較複雜，它可以讓用家選擇載著原程式的磁碟機，如 B：磁碟機，HEX 或 PRN 檔案的磁碟機儲存地，或者命令 ASM.COM 不用產生 HEX 及 PRN 的檔案。在 PA— RMS 中包括了三個字母，我們叫它爲 SHP。"S" 爲載著原程式的磁碟機，如A至F，"H" 爲說明那一個磁碟機儲存 HEX 檔案，"P" 爲說明那一個磁碟機儲存 PRN 檔案。簡表 7.2說明了它們的關係。

S	H	P	說 明
B	A	A	原程式在B：磁碟機， HEX 及 PRN 檔案在 A：磁碟機
B	Z	A	原程式在B：磁碟機，不用產生 HEX 檔案，PRN 檔案儲存存在 A：磁碟機
B	Z	X	同上，惟 PRN 檔案在螢幕顯示
B	B	Z	原程式在 B：磁碟機，HEX 檔案在 B：磁碟機儲存，不需產生 PRN 檔案
B	Z	Z	原程式在 B：磁碟機，不用產生 HEX 及 PRN 檔案

（表 7.2）

2、載入（ LOAD ）

當 HEX 檔案產生後，用家可以使它變爲以 COM 爲類別的工具程式，這便要用到載入（ LOAD ）這個步驟了。其過程如圖 8.1 所示；

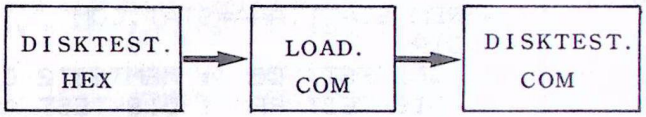


圖 8.1

LOAD 本身已經是一個工具程式，而被載入的檔案的類別一定要HEX 。它的格式爲；

（A）LOAD DISKTEST

（B）LOAD B:DISKTEST

（A）的格式爲從一早設定的磁碟機中，載入一個叫做 DISKTEST.HEX的檔案，而其所產生的 DISKTEST. COM 爲一個工具程式，要執行這個程式，用家就需打入 DISKTEST 便可。（B）的格式爲用家可以說明在那一個磁碟機存取檔案，而其產生的結果與（A）格式相同。有一點要注意的，就是 HEX 檔案的起始地址一定要 100 H ，換句話說，就是 TPA 的開始地址，而 HEX 檔案內的地址一定要由細到大的順序編排。假如地址與地址之間有空隙，LOAD 便用 0 來填滿它。假如 HEX 檔案的起始碼不是 100 H ，那便要用除錯封裝（ DDT ）來載入了。

（未完，下期續）

```
A> ERA USELESS.COM
```

例 1.1

```
A>ERA A*.COM
```

例 1.2

```
A>
A>ERA *.*
All (y/n)?Y
A>DIR
No file
```

例 1.3

```
A>DIR B:D*.**
B: DISKTEST COM : DISKTEST HEX : DISKTEST PRN : DISKTEST ASM
```

例 2.2

例 2.1

```
A>DIR
A: CPUTEST  COM : MEMTEST2 COM : DISKTEST COM : DISKTEST HEX
A: DISKTEST PRN : DISKTEST ASM : Z80ASM   COM : FILECOPY COM
A: COMPARE  COM : ASMXREF  COM : ASCDUMP  COM : SCRAMBLE DOC
A: SCRAMBLE ASM : ASMXREF  ASM : ED       COM : PIP      COM
A: STAT     COM : FORMAT   COM : CAT      COM : BOOT     COM
A: AUTORUN  COM : MFT      COM : PATCH    COM : ASM      COM
A: COPY     COM
```

例 3.1

```
A>
A>REN XYZ.COM=PIP.COM
A>DIR
A: CPUTEST  COM : MEMTEST2 COM : DISKTEST COM : DISKTEST HEX
A: DISKTEST PRN : DISKTEST ASM : Z80ASM   COM : FILECOPY COM
A: COMPARE  COM : ASMXREF  COM : ASCDUMP  COM : SCRAMBLE DOC
A: SCRAMBLE ASM : ASMXREF  ASM : ED       COM : XYZ      COM
A: STAT     COM : FORMAT   COM : CAT      COM : BOOT     COM
A: AUTORUN  COM : MFT      COM : PATCH    COM : ASM      COM
A: COPY     COM
```

例 3.2

```
A>REN STATT.AAA=STAT.COM
A>DIR
A: CPUTEST  COM : MEMTEST2 COM : DISKTEST COM : DISKTEST HEX
A: DISKTEST PRN : DISKTEST ASM : Z80ASM   COM : FILECOPY COM
A: COMPARE  COM : ASMXREF  COM : ASCDUMP  COM : SCRAMBLE DOC
A: SCRAMBLE ASM : ASMXREF  ASM : ED       COM : XYZ      COM
A: STATT    AAA : FORMAT   COM : CAT      COM : BOOT     COM
A: AUTORUN  COM : MFT      COM : PATCH    COM : ASM      COM
A: COPY     COM

A>REN ABC.COM=XYZ.COM
```

例 3.3

```
A>REN ABC.COM=XYZ.COM
File exists
```

例 3.4

```
A>REN AAA.COM=XXX.COM
No file
```

例 5.1

```
A>SAVE 2 QQ.WW
A>DIR QQ.*
A: CPUTEST  COM : MEMTEST2 COM : DISKTEST COM : DISKTEST HEX
A: DISKTEST PRN : DISKTEST ASM : QQ       WW  : FILECOPY COM
A: COMPARE  COM : ASMXREF  COM : ASCDUMP  COM : SCRAMBLE DOC
A: SCRAMBLE ASM : ASMXREF  ASM : ED       COM : ABC      COM
A: STATT    AAA : FORMAT   COM : CAT      COM : BOOT     COM
A: AUTORUN  COM : MFT      COM : PATCH    COM : ASM      COM
A: XX       QQ  : COPY     COM
A>
A>DIR
A: CPUTEST  COM : MEMTEST2 COM : DISKTEST COM : DISKTEST HEX
A: DISKTEST PRN : DISKTEST ASM : QQ       COM : FILECOPY COM
A: COMPARE  COM : ASMXREF  COM : ASCDUMP  COM : SCRAMBLE DOC
A: SCRAMBLE ASM : ASMXREF  ASM : ED       COM : ABC      COM
A: STATT    AAA : FORMAT   COM : CAT      COM : BOOT     COM
A: AUTORUN  COM : MFT      COM : PATCH    COM : ASM      COM
A: XX       QQ  : COPY     COM
A>
```

例 4.1

CTRL-P

```
A>TYPE SCRAMBLE.ASM
;
;Scramble.asm - program to scramble CP/M files
;using an 8 byte password
;
month   equ     3       ;last..
day     equ     14      ;..modification..
year    equ     79      ;..date
;
;Scrambling is done in place, i.e. the file is
;modified on top of itself.  The same password
;used to scramble the file is used to unscramble
;it, using the exact same command.  This is because
;the scrambling code is exclusive-or'ed with the
;data file, and two same exclusive ors result
;in the original value being returned.
;
;Command format:
;
;       scramble filename.type password
;
;where password is any 8 character string which
;is allowable as a file name (i.e. no '.', etc).
;
MF      SET     0       ;SHOW MOVE NOT REQUESTED
CF      SET     0       ;SHOW COMP NOT REQUESTED
;
;(FROM EQU8.LIB...)
;DEFINE SOME MACROS TO MAKE THINGS EASIER
;
;DEFINE DATA MOVE MACRO: MOVE from,to,length
;       from may be addr, or quoted string
;
MOVE    MACRO   ?F,?T,?L
        IF      NOT NUL ?F
        IRPC    ?C,?F
?Q      SET     '&?C&?C' ;;TEST FOR QUOTE
        EXITM
        ENDM
        IF      ?Q EQ ''''
        LOCAL   ?B,?Z
        CALL    ?Z
?B      DB      ?F
?Z      POP     H       ;GET FROM
        LXI     B,?Z-?B ;GET LEN
        ELSE
        LXI     H,?F
        ENDIF
        ENDIF
        IF      NOT NUL ?T
        LXI     D,?T
        ENDIF
        IF      NOT NUL ?L
        LXI     B,?L
        ENDIF
        CALL    MOVER
MF      SET     -1      ;;SHOW EXPANSION
        ENDM
```

CTRL-P

例 5.2

```
A>SAVE 40 B:SS.WW

A>DIR B:S*.*
B: SS       WW  : SCRAMBLE DOC : SCRAMBLE ASM : STATT     AAA
```

PRODOS 縱橫

普魯斯

第2篇：磁碟格式

PRODOS 的面世，使 APPLE Ⅱ＋及 Ⅱe的用家受惠不淺，基本上的評價是快速而具結構性，當然我們亦付出了代價，例如將 16 K RAM 咭保留予 PRODOS 使用，令 INTEGER BASIC 無法運用。其次 DOS 3.3 系統，搬移至 RAM 咭上以騰出 10.5 K空間這種好處，基於以上原因亦被迫放棄。但它的確十分好，因此我預備和各位探討 PRODOS 磁碟一些技術問題。

在 PRODOS格式化後的磁碟上，它明顯地與 DOS 3.3的有所不同，它將560 個 SECTOR改變爲 280 個邏輯方塊（ LOGICAL BLOCK），並且將會佔用 6 個BLOCKS，在 TECHNICAL MANUAL上指出在第 0 及第 1 個 BLOCK 上是存放載入程式的代碼，而 BLOCK 2 是選擇性代碼， BLOCK 2－5 存放 VOLUME DIRECTORY 之用，它包括了一個指向器來指向一個 BLOCK 之用。

在 BLOCK 6 上是 VOLUME BIT MAP 的所在地，這相當昔日的 VTOC ，在裏面的每一個 BIT 代表磁碟上的一個 BLOCK ，因此某個 BIT 的值等於 1 的話即該相對位置的 BLOCK 正被空置來用，等於零代表已被使用。

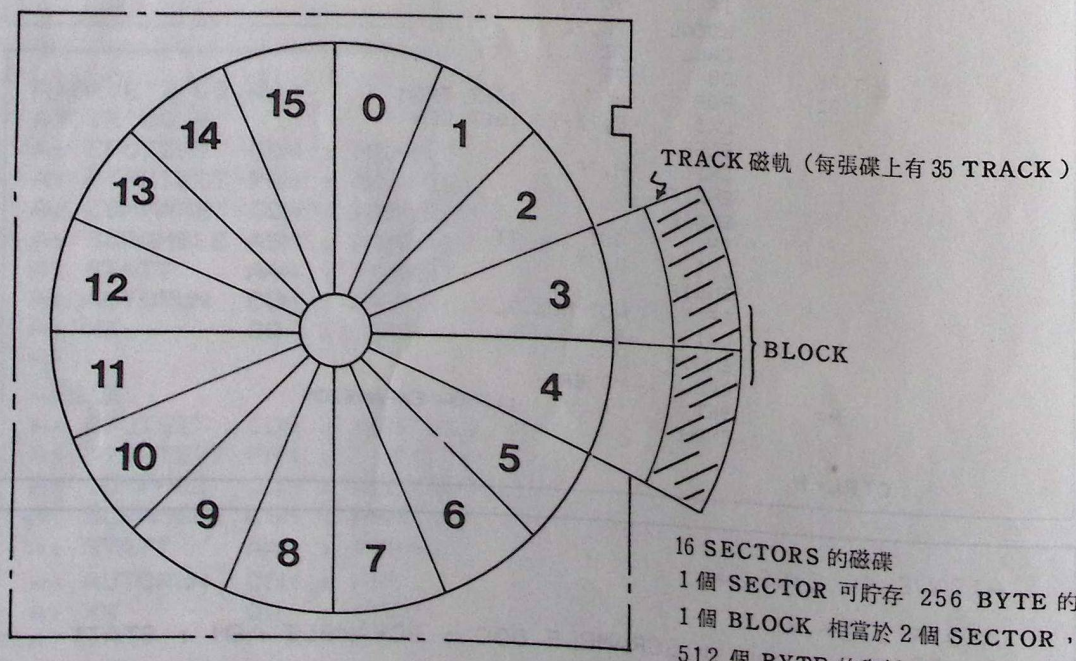
由於 BLOCK 6有4096個BITS（512BYTE×8＝4096 BITS），因此它可以代表 4096 個 LOCK 的狀態，但事實上對 5 ¼吋磁碟來說只要35個 BYTE（ 35×8＝280 BITS）就足夠了，但對硬碟系統來說， 32 M BYTE 的貯存量須要16個 BLOCKS才足夠。

當一個檔案被存放進入新格式化磁碟時，會灌錄在緊隨 VOLUME BIT MAP 所佔用的 BLOCK 後，這對 5 ¼吋磁碟來說是 BLOCK 7 ，其次一個作爲指引的檔案佔用其中的 512 BYTE ，即一個 BLOCK ，而內含數據的檔案則佔用 513個 BYTES 至最大 128 K BYTES 不等（ 5 ¼吋磁碟最大只能載錄 128 K 的檔案）。因此每存入檔案最少要佔用 2 個 BLOCKS。其中這些檔案的結構是樹狀延伸的，對於使用方面會很具結構性。

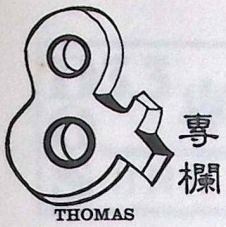
當檔案長度大於 128 K 的時候會以樹狀方式存放，極限是16 M BYTES， 不過 5 ¼吋磁碟不能享受這個好處了。

在下期我將與各位訊論 PRODOS BOOT碟的過程，再會！ □

已被格式化的 PRODOS磁碟結構



16 SECTORS 的磁碟
1 個 SECTOR 可貯存 256 BYTE 的數據
1 個 BLOCK 相當於 2 個 SECTOR，即可貯存 512 個 BYTE 的資料。



AMPER-CAT 及 AMPER-FREECAT

(一) 序言

當程式在 APPLE SOFT 的環境中呼叫 6502 機械語言時，最爲用家常用的爲 " CALL " 但除了 CALL 之外，APPLESOFT 還提供了兩個命令，它們分別爲 " USR ( ) " 及 " & "。筆者不打算在此論述 " CALL " ，因坊間已有很多書籍及實用的程式供參考及應用。 " USR " 爲另一個呼叫機械語言的命令，它的實際操作情形在此作一簡略的說明，設定 A = USR (X*Y) ，它把 (X*Y) 這個論據放在 FAC (FLOATING POINT ACCUMUL—ATOR) 中，然後跳至地址碼 $0A 中，在這地址開始，用家放進了一個 " JSR " 的命令，引導它跳至用家要執行的程式，當此程式遇到 " RTS " 時，它便跳回 APPLE SOFT 中。

舉例如下：

```
]CALL-151
*000A:4C 00 03          JSR $0300
*0300:60                RTS
<CTRL-C><RETURN>
]10  A=USR(8*9)
]20  PRINT A
]RUN
 72
```

各位現在應已有些印像，但使用這個命令時，用家一定要對整個程式的流程有深入的了解，才可以準確地運用自如。還有一個命令可以取代它的，就是 "&" 了，比較起來 " & " 比 " USR " 執行還要快及好。

(二) 奇妙的 "&"

從 APPLE SOFT 呼叫機械語言，要算是最快的，莫如這可愛的 "&" 了。但在 APPLE Ⅱ的參考書籍中却鮮有提及，似乎 APPLE Ⅱ的發明者把它遺留了，有這個好的命令却沒有文獻支持，實在可惜，故筆者打算在此借一個小小的篇幅，一齊與讀者去探索這個豐富的寶藏。它的流程可在此作一略述。當 APPLE SOFT 遇到 "&" 時，它便會立即跳至 RAM 的地址 $3F5 中，我們叫這個地址爲 "& 向量" (AMPERSAND VECTOR)。在 $3F5 中，用家可以放進子程式的 " JSR " 地址碼，而 APPLE 這時便自動跳至用家所說明的程式並即時執行，當它遇到 " RTS " 的指令時，操作終止而跳回 APPLE SOFT 中。讀者們也許會問， "&" 的操作與 " USR " 差不多，但爲甚麼 "&" 是最快的？這便牽涉到 APPLE 在執行命令時的流程了。當 APPLE 在執行一個命令時，如 " CAL—L " ， " USR " 及 "&" 等，它都是先在命令表中核對，核對無誤後，再找這個命令的起始地址表，找到了後，跳至所屬的地址中， " CALL " 及 " USR " 還要對程式中交來的論據 (ARGUMENT) 加以解釋才能相應執行，但 " & " 則不然，因爲它永遠跳至 $3F5 的地址中執行任務，所以是最直接的，記著，其他的命令是 APPLE SOFT 加以翻釋才執行，而 "&" 則不然，故此愈多直接的命令，程式執行速度愈快。如能善用它，用家可以自己開發更多更快的新命令。

```
1   REM  ************************
2   REM  *                      *
3   REM  *       AMPER-CAT      *
4   REM  *                      *
5   REM  *  (C) COMPUHOUSE 1984 *
6   REM  *                      *
7   REM  ************************
15   POKE 1013,76: POKE 1014,110:
     POKE 1015,165
```

例子一

(三) 實驗

在說過了 "&" 的流程操作後，我們且來做一個小小的實驗程式。讀者中可有因要時常鍵入 " CATALOG " 看磁碟的檔案？又每次都沒有打錯一個字母？筆者就因工作的需要而時常鍵入 CATALOG ，但打錯了又要重頭來，雖然市面上有很多 DOS 的命令改錯程式，但它們的長度與本程式比較起來却長得嚇人，而且要時常做相同的命令却也眞惱人。本程式就利用了 "&" 去取代 " CATALOG " ，而且可以加入現有的 " HELLO " 程式中，每次 BOOT 起 DOS 後，用家只要打進 "&" 便可以做到這個效果，程式見例一。它只是做了一個簡單的 " JSR " 指令，跳至 CATAL—OG 的進入點 (ENTRY POINT) $A56E 中。其效果可見例子二，這是一個很簡潔的應用例子。

筆者因工作上的需要，每日處理大量的檔案記錄工作，要與一張磁碟加以題目說明及磁碟扇區的剩餘量，當利用例子一的程式如：

" & COMPUHOUSE VOL＃1 "

它做了 " CATALOG " 的工作後，便出現了 " SY—NTAX ERROR " 的句語，這樣做殊不美觀，試想想，整份報表紙都有這句句語，極不好看，於是筆者再寫了例子三

的程式。

程式在第三頁區開始$0300它首先觸發了"&向量—$3F5"在$3F5跳至CATALOG的進入點$A56E中，當遇到"RTS"而折回時，再叫它跳至DOS的輸入／輸出表（IOB—$03E3），讀出第17磁軌0號扇區（VTOC）的資料，把這些資料讀進鍵盤輸入緩衝區（KE—YBOARD BUFFER—$200）中，計算一下扇區的剩餘量，再呼叫APPLE SOFT的幾個副程式，如CRDO—$DAFB，DATA—$D995，PRBLNK—$F948，LINPRT—$ED24，STROUT—$DB3A等而組成，這程式可以在"&"之後鍵入任何字串，而APPLE不會有"SYNTAX ERROR"的反應，因爲筆者用了一個APPLE SOFT的副程式— DATA，在程式來說，它只是把用家鍵入的字串作資料看待，故不會報以"SYNTAX ERROR"。

這個程式除了可以令用家鍵入任何字串作爲磁碟的標題外，它還告訴用家磁碟扇區剩餘的數量，當要用作印表機輸出時，可先鍵入"PR#1"。例子四爲這程式執行後的效果。

關於"&"的應用及程式，下期再與各位討論。□

例子三

```
JBLOAD AMPER-FREECAT
JCALL-151

*300.3BCL

0300- A9 0B 8D F6 03 A9 03 8D
0308- F7 03 60 48 98 48 8A 48
0310- 20 6E A5 20 E3 03 84 F9
0318- 85 FA A0 03 A9 00 91 F9
0320- A0 04 A9 11 91 F9 A0 05
0328- A9 00 91 F9 A0 08 A9 00
0330- 91 F9 C8 A9 02 91 F9 A0
0338- 0C A9 01 91 F9 A4 F9 A5
0340- FA 20 D9 03 A0 0D B1 F9
0348- F0 02 D0 54 A9 00 8D 90
0350- 03 8D 91 03 A0 38 B9 00
0358- 02 A2 07 0A 90 08 EE 91
0360- 03 D0 03 EE 90 03 CA 10
0368- F2 C8 C0 C4 90 E8 20 FB
0370- DA 20 48 F9 AD 90 03 AE
0378- 91 03 20 24 ED A9 92 A0
0380- 03 20 3A DB 20 FB DA 20
0388- 95 D9 68 AA 68 A8 68 60
0390- 00 00 20 46 52 45 45 20
0398- 53 45 43 54 4F 52 53 00
03A0- A9 A8 A0 03 20 3A DB 60
03A8- 20 45 52 52 4F 52 20 52
03B0- 45 41 44 49 4E 47 20 56
03B8- 54 4F 43 00 2A
*
```

例子二

```
]&

DISK VOLUME 254

*A 051 APPLE CALENDAR
*A 003 APPLE CALENDAR & SUM
*A 003 APPLESOFT TO FORTRAN
*A 020 BEND OF ROUNDING
*B 003 BOOLEAN TUTORIAL
*A 004 BOOLEAN TUTORIAL-MAIN PROGRAM
*T 004 CAL.FILE-1983
*T 004 CAL.FILE-1984
*T 004 CAL.FILE-1985
*T 004 CAL.FILE-1986
*A 036 COMPUTEXT
*B 002 DVORAK KEYBOARD
*A 008 DYNAMIC CALENDER
*A 010 GAME OF CONNECTING FOUR
*A 009 GAME OF MATHEMATIC
*A 027 GAME OF MAZING
*A 018 GAME OF SPELLING TEST
*A 006 GREETING
*A 009 GREETING TRANSFER
*B 006 MAZE SHAPES
*B 002 PEELING LISTING 5
*A 006 PEELING THE TEXT SCREEN
*A 005 RETRIEVE THE DELETED PROGRAM
*B 034 SUM1
*T 002 TEXT TEST
 A 023 GAME OF SABOTAGE
 T 002 HISAB

   197 FREE SECTORS
```

例子四

```
]& COMPUTING-AGE DISK.2

DISK VOLUME 254

 B 003 BPHTCH
 B 010 CHAR1
 A 018 CONTAINER HANDLING
 A 019 CRT COLOR DESMONTRATION TEST
 A 020 GAME OF CRYPTO
*A 023 GAME OF SABOTAGE
*A 000 GREETING
 A 012 HIGH SCORE KEEPER
 A 002 LEAD
 T 002 LOADER
 A 003 LOADSUM2
 A 006 LOWER CASE CHIP CONVERTER
 A 045 MATH TOOL KIT 1
 A 012 PATCH2
 A 038 POKER
 A 002 POKER SHOWHAND
 A 012 RETRIEVE FONTASTIC
 A 006 SAFELY HANDLING PROGRAM
 B 002 SCROLL!!
 B 034 SUM2
 A 027 SUPERFONT
 A 045 Z
 T 007 FIDFILE

   148 FREE SECTORS
```

DOS 3.3 專欄

甘貴祥

指令處理器的工作原理

用滙編語言直接控制DOS指令

對於不用BASIC及DOS直接指令，而能在滙編語言程式中可以控制 RUN 或 BRUN 磁碟檔案的原理大家是否明白？要尋出其中原理，必須先要了解一些有關的指令處理器（COMMAND HANDLERS）。

蘋果的磁碟操作系統（DOS）容許電腦用家從磁碟機輸入資料至電腦的記憶，或相反，從電腦記憶將資料存入磁碟機。相信每一位擁有磁碟機的讀者都會知道，執行這些輸出／輸入操作的子程式有多個不同的名稱：BRUN，BLOAD，RUN，SAVE 等等。它們冠稱之爲 DOS 語言。

COMMAND HANDLER（以下簡稱 COM HAND）是透過一個 COM HAND 輸入點表（ENTRY POINT TABLE）而呼喚相若的子程式。

每個子程式都有一個固定的地址或開始的輸入點。要在滙編語言程式中使用這些子程式只須要 JUMP 至該指定的地址即可。至於如何 JUMP 及所 JUMP 的地址是從那點開始，則可參考輸入點表。該表如下：

INIT	$A54F
LOAD	$A413
SAVE	$A397
RUN	$A4D1
CHAIN	$A4F0
DELETE	$A263
LOCK	$A271
UNLOCK	$A275
CLOSE	$A2EA
READ	$A51B
EXEC	$A5C6
WRITE	$A510
POSITION	$A5DD
OPEN	$A2A3
APPEND	$A298
RENAME	$A281
CATALOG	$A56E
MON	$A233
NOMON	$A23D
PR#	$A229
IN#	$A22E
MAXFILES	$A251
FP	$A57A
INT	$A59E
BSAVE	$A331
BLOAD	$A35D
BRUN	$A38E
VERIFY	$A27D

首先進入監督程式（CALL—151）。按入 A56 EG（緊記必須載入 DOS 於記憶內，及插放在磁碟機中的磁碟能有效地 CATALOG）。

參看 COM HAND 輸入點表，先前的輸入數值是直接命令電腦 CATALOG 磁碟。大家應該明白G是代表執行在G之前的四位數值，地址開始的子程式。

假如要 LIST CATALOG 這子程式的列表，可以輸入 A56 EL，L在監督程式中代表 LIST。

某些子程式可以單獨使用，但一些則要連同一些資料，例如檔案名稱，一起使用。BRUN 就是其中一個例子。

取一隻空碟，INIT 後，從監督程式中輸入下列短短的一句程式：

```
*800.80E

0800- A2 00 E8 8A 9D D0 07 E0
0808- 1A D0 F7 60 00 00 00
```

程式全是「B」檔案（即二進制碼檔案）。按入 800L，及按 RETURN，程式會如下顯示：

```
0800-   A2 00      LDX   #$00
0802-   E8         INX
0803-   8A         TXA
0804-   9D D0 07   STA   $07D0,X
0807-   E0 1A      CPX   #$1A
0809-   D0 F7      BNE   $0802
080B-   60         RTS
080C-   00         BRK
080D-   00         BRK
080E-   00         BRK
080F-   FF         ???
0810-   FF         ???
0811-   FF         ???
0812-   FF         ???
0813-   FF         ???
0814-   FF         ???
0815-   FF         ???
0816-   FF         ???
0817-   FF         ???
0818-   FF         ???
```

假如懂得滙編語言，應該知道上述一句指令是有甚麼用。假如不知道，試按 800G 看看有些甚麼出現。

以 BSAVE ABC，A$800，L$0E 將上述程式存入磁碟。

指令處理器的工作原理

從滙編語言程式中，BRUN 一個檔案需要兩件事。首先要有個檔案名字，它必須存於 DOS 能找到的地方，這個地方被稱爲 PRIMARY FILE NAME BUFFER，它由記憶位址 $AA75 開始。

其次，需要完成一個 JUMP 指令(JMR)或JUMP到一個子程式指令（JSR），此指令必須要有輸入點地址。

試打入下列一個機械語言程式：

```
0300-   A2 00      LDX   #$00
0302-   A9 A0      LDA   #$A0
0304-   9D 75 AA   STA   $AA75,X
0307-   E8         INX
0308-   E0 1E      CPX   #$1E
030A-   D0 F6      BNE   $0302
030C-   A9 C1      LDA   #$C1
030E-   8D 75 AA   STA   $AA75
0311-   A9 C2      LDA   #$C2
0313-   8D 76 AA   STA   $AA76
0316-   A9 C3      LDA   #$C3
0318-   8D 77 AA   STA   $AA77
031B-   4C 3E A3   JMP   $A33E
```

按 300 L ，研究這些反組合語言程式列表。位置 300 至 30 A 儲存 A0入記憶位址 $AA 75至 $AA 92，並清除 PRIMARY FILE NAME BUFFER。假如在輸入機械語言程式時，不先將 BUFFER 全部 CLEAR，很容易會盡花新程式，特別如果在較早之時存有大型程式的機械語言檔案不 CLEAR 緩衝器，可能導致錯誤發生。

30C－30E將 C1 數值存入緩衝器（$AA75）的首個記憶地址。C2，C3 數值同樣被存入第二及第三的緩衝器位址（AA 76－AA 77）。C1，C2及C3 是A、B 及C的十六進制代表——並成爲二進制檔案的檔案名字（即ABC）。最後，31B JUMP 到 COM HAND的進入點準備 BRUN 。

在未 BRUN 之前， BSAVE 這程式（用 BSAVE DEMO，A$300，L$20。

當兩個檔案都 SAVE 入磁碟後，可以試從監督程式中 RUN 後者程式（打入 300 G），然後又再嘗試用直接磁碟指令（ BRUN DEMO）。

DEMO 程式會 CLEAR PRIMARY FILE NA—ME 緩衝器，載入新檔案準備 BRUN ABC ，然後 J—UMP 到 BRUN 子程式的輸入點。

嘗試將 DEMO 程式 31B 行作下列多項不同的修改，然後利用G指令 RUN 每一個新的DEMO 程式。

31B：4C 71 A2（LOCK ABC 檔案）
31B：4C 75 A2（UNLOCK ABC檔案）
31B：4C 5D A3（BLOAD ABC 檔案）
31B：4C 7D A2（VERIFY ABC檔案）
31B：4C 63 A2（DELETE ABC檔案）

同樣，對 APPLESOFT 的檔案亦可如此做。

任何人想令DOS指令不能正常工作，辦法及步驟相當簡單。只要將數值 60存入 COM HAND進入點表所陳述的該指令的位址即可。60 是滙編語言指令，代表從子程式中 RETURN（RTS）。舉例說，要廢止 CATALOG指令，輸入 A56E：60亦可。□

（上接 P.61 ）

第一組	第二組	第三組	第四組	
00	00	10	00	（位元）
向上	向下	向左	向右	（方向）

你可以知道第一，第二和第四組的位元值都是 00，表示這些方向物體都是靜止不動，因爲移動距離是 0。第三組的移動距離是 10 以十進位表示便是 2，所以物體便以每次兩 BIT 的距離向左移動了。現在要改變物體移動速度及方向不是一件很容易的事嗎？假如你想物體改向右方以兩BIT移動，可以把第三組改爲 00 使它向左靜止不動；而把第四組改爲 10 使它向右以兩 BIT 移動。經過這樣的更改，整個位元組的值便變成：

00000010(二進位) ＝ $02

換言之，把 $02 這個數值存入 PATH 的位址內便成了。再看程式 LISTING＃2 行號 40 和 50，這就是我們把 2 和 8 在測試邊界時存入 35520 （路綫表位址）的原因了。假如還有不明白的話，再舉例看看倘若我們想物體以 90 度角向右上角以 3BIT 的速度移動，應該如何更改移動單位的值。

向右上角移動物體必須同時向上及向右方以 3 BIT 移動（相信還記得物理學的 RESULANT VELOCITY＝X VELOCITY＋Y VELOCITY 原理吧）。於是我們把第一組位元改爲 11，而第四組位元改爲 11，其餘仍然是 00 靜止不動。整個位元組的便變成：

11000011(二進位) ＝ $C3

所以只要把 $ C3或由 BASIC 存入195這個數值在路綫表位址，便使物體向右上角以3 BIT 移動。你現在是否可以從 RBOMB PATH 1，PTH 那條路綫表找出它的速度和方向呢？

最後要補充的是，如果我們想物體在某些移動數步（MOVE）靜止不動，例如希望怪獸在畫面上有時靜止，但有時移動的不規則路綫，原則上每組位元都改爲 00 不就可以嗎？但將請記着位元組 00是路綫完結的旗號，所以只能把位元組值改爲 FF以表示物體不移動了。

現在就試試把已造好的 ANIMATION ROUTINE 用 BASIC 來更改以上所談的各種位址，看看效果如何！

下期再談有關碰撞檢知，背景及聲音等各種效果，並且再列出程式第二部份的組合語言程式。再見！□

編印單個LABEL

許多MAILING LIST 程式都能幫助我們處理大量的郵寄名單，如要編印數百個或數千個的LABEL，這類商用軟件會非常有用。可是，假如我們只是要編印一至二個的LABEL，則未免大材小用。我們在此提供一個短短的APPLESOFT 程式供讀者作此用途。程式可讓閣下選擇編印形式，而且在編印之前，可於螢幕上先檢查清楚有否錯誤然後才輸出往印字機。

這個程式特別適宜編印一個至數個的地址LABEL、檔案標記、磁碟LABEL等等。程式是爲ESPON MX—80或相融系列印字機而寫。有三種字款選擇：標準字（10CPI），壓縮字（16、5CPI）或雙倍濶度（9CPI），每行長度字數依次爲25、45、21字。適宜使用3吋長，7／8吋濶的LABEL紙。

程式使用非常簡單，開始時只要指明要編印多少個LABEL，每個LABEL的行數。然後選擇需要的字體。

用家此時可輸入資料（如有錯誤可以更改）。當螢幕上所顯示的東西全部無誤，印字機可以隨時印出。

由於ESPON系列或相融的機種是採用ESC 來控制字款式樣，所以程式內用到ESC字符（CHR$（27））。變數LT調定印字模式，DW是印雙倍濶度字樣，SP設定行數數目。使用其他ESPON以外的讀者，可改變這幾個變數來配合本身的機種。（參考行句160～180）。

行句190至230 建立一個輸入子程式。變數N是指定每個LABEL編印的行數。

當然讀者們可自行改變LABEL紙的呎吋及編印字款模式以方便自己的需要。辦法相當簡單，只要改變SP（行長度），字體款式（LT及DW）。或者將行號80、120 的〔DIM L$（5）〕及〔OR N＞5〕中的5改爲自己喜歡的行數。如要遷就每個LABEL與LABEL之間的距離，可以於行句300中最後一段〔NEXT I:PRINT:NEXT C〕中之PRINT之後多加PRINT字句即可，每個LABEL與LABEL之間的距離即可決定。

本程式磁碟中的FILENAME是PRINTER LABEL。□

```
10  REM  ******************
20  REM  *                *
30  REM  * LABEL PRINTER *
40  REM  *                *
50  REM  ******************
80  TEXT : HOME : CLEAR : POKE 34
    ,10: DIM L$(5)
90  INVERSE : PRINT "***   L A B
    E L     P R I N T E R   ***"
    : NORMAL
100 VTAB 5: INPUT "HOW MANY LABE
    LS TO PRINT?      ";NN
110 VTAB 6: INPUT "HOW MANY LINE
    S PER LABEL?      ";N
120 IF N <  = 0 OR N > 5 THEN  GOTO
    110
130 VTAB 8: PRINT "TYPE STYLE.
    1)STANDARD": PRINT  TAB( 14)
    "2)CONDENSED": PRINT  TAB( 1
    4)"3)DOUBLE WIDTH ";: INPUT
    "";TY$
140 IF TY$ = "1" OR TY$ = "2" OR
    TY$ = "3" GOTO 160
150 GOTO 130
160 IF TY$ = "1" THEN LT = 18:SP
     = 25:DW = 18
170 IF TY$ = "2" THEN LT = 15:SP
     = 45:DW = 18
180 IF TY$ = "3" THEN LT = 14:SP
     = 21:DW = 15
190 FOR NL = 1 TO N
200 VTAB 12: PRINT : PRINT "LINE
     #";NL;":": PRINT : PRINT  CHR$
    (91);: FOR L = 1 TO SP: PRINT
    ".";: NEXT L: PRINT  CHR$ (9
    3)
210 VTAB 15: HTAB 2: INPUT "";L$
    (NL)
220 IF  LEN (L$(NL)) > SP THEN  HOME
    : GOTO 200
230 NEXT NL
240 HOME : VTAB 14: PRINT "THIS
    IS THE WAY THE LABEL WILL LO
    OK:": PRINT
250 INVERSE : FOR C = 1 TO 5: PRINT
    L$(C): NEXT C: NORMAL
260 VTAB 22: PRINT "IS THIS CORR
    ECT? (Y/N) ";: GET AN$
270 IF AN$ = "Y" THEN  POKE 34,3
    : GOTO 290
280 IF AN$ = "N" THEN  HOME : GOTO
    190
290 HOME : PRINT : PRINT  CHR$ (
    4)"PR#1"
300 FOR C = 1 TO NN: FOR I = 1 TO
    5: PRINT  CHR$ (DW); CHR$ (L
    T);L$(I): NEXT I: PRINT : NEXT
    C
310 PRINT  CHR$ (4)"PR#0": POKE
    34,3: HOME : VTAB 5: PRINT "
    RUN MORE LABELS? (Y/N) ";: GET
    AN$: PRINT AN$
320 IF AN$ = "Y" THEN  GOTO 80
330 POKE 34,0: HOME : END
```

高解像頁處理工具箱

在第七期裡我們刊登了一個快速載入高解像圖形的工具程式，相信讀者會發現它是相當實用。不過今期，筆者要爲大家提供一個更加實用的工具程式——高解像頁處理工具箱。其功能較第七期所刊出的強效得多。

這個高解像頁處理工具箱程式所能做的包括如下：

（1）對調 PAGE 1 及 PAGE 2。
（2）將 PAGE 2 的圖像加蓋在 PAGE 1 之上。
（3）將 PAGE 1 的影像反相印出。
（4）顯示 PAGE 1
（5）顯示 PAGE 2
（6）載入高解像圖頁。

本程式同有二個程式列表：（1）HI－RES TOO－LKIT 及 TOOL。

前者是 APPLESOFT 程式，而後者是機械語碼。

不過程式磁碟的用家要留意，本軟件在磁碟上只以一個 FILE 儲存，FILENAME是 HI－RES TOOLKI－T，它是一個長12個 SECTOR的BINARY FILE。運行時，只需直接 BRUN HI－RES TOOLKIT 即可。

```
10  REM     ********************
20  REM     *                  *
30  REM     *  HI-RESTOOL KIT  *
40  REM     *                  *
50  REM     ********************
55  POKE  - 16297,0: POKE  - 1630
    2,0
60 SWITCH = 2054:SUPERIMP = 2112:
    INVERT = 2176
70  TEXT : HOME : PRINT : INVERSE

80  HTAB 13: PRINT "HI-RES TOOLKI
    T"
90  NORMAL : PRINT : HTAB 13: PRINT
    "-BY COMPUHOUSE-"
100  GOSUB 630: GOSUB 630: PRINT

110  PRINT : INPUT "INSTRUCTIONS
    (Y/N)?";A$: IF A$ = "N" THEN
    120
115  GOSUB 680
120  TEXT : HOME : PRINT : INVERSE

130  HTAB 13: PRINT "HI-RES TOOLK
    IT": NORMAL
140  VTAB 8: HTAB 17: PRINT "=MEN
    U="
150  PRINT
160  PRINT : PRINT "1)SWITCH PAGE
    S"
170  PRINT "2)SUPERIMPOSE PAGES"
```

話說回頭，假如讀者自己打入此兩個程式列表，必須留意，筆者在此程式中採用了一種新技巧，將 APPLESOFT 程式混同機械語言一併以BFILE方式儲入磁碟上，說得清楚一點，是將一個機械語言程式放在一個 APPLESOFT 程式之前，並將之作爲一個大"機械語言" 程式 BSAVE 起來。筆者將於下期詳細講述這種新技術。有興趣研究這種新儲錄 FILE 技巧的讀者可要留意。

讀者在打入程式（1）HI－RES TOOLKIT 程式之前，要先加多一行 LINE 0 之後加上 REM，並且按入189個破折號〔—〕，然後便打入程式。跟著打入 TOOL，由位址 $800 開始。在 RUN 程式之前，必須遵照下列方法去做，否則程式如不能運行筆者恕不負責：

方法如下：RUN 780

然後螢幕會顯示一個 " CHANGE ROUTINE"，照螢幕指示去做，BSAVE "HI－RES TOOLKIT" 入磁碟。之後，兩個程式會合而爲一，只要按 BRUN HI－RES TOOLKIT 即可。新的程式會是一個〔機械語言〕程式。□

```
180  PRINT "3)INVERT PAGE ONE"
190  PRINT "4)VIEW PAGE ONE"
200  PRINT "5)VIEW PAGE TWO"
210  PRINT "6)BLOAD A PICTURE": PRINT
    "7)QUIT"
220  PRINT : INPUT "NUMBER SELECT
     ON: ";A$:N =  VAL (A$): IF
    N < 0 OR N > 255 THEN 120
230  ON N GOTO 280,280,280,360,36
    0,420,520
240  GOTO 120
260  REM     ** SELECTION 1,2 OR 3
     **
280  POKE  - 16300,0: POKE  - 163
    04,0: GOSUB 630
290  ON N GOTO 300,310,320
300  CALL SWITCH: GOSUB 630
305  GOTO 120
310  CALL SUPERIMP: GOSUB 630
315  GOTO 120
320  CALL INVERT: GOSUB 630
325  GOTO 120
360  REM    ** VIEW PAGE 1 OR 2 **

380 PG = N - 3
390  POKE  - 16302,0: POKE  - 162
    97,0: POKE  - 16304,0: POKE
    ( - 16301 + PG),0
400  GET A$: PRINT : GOTO 120
420  REM    ** BLOAD A PICTURE**
440  HOME : PRINT : INPUT "NAME O
```

高解像頁處理工具箱

```
     F PICTURE: ";A$: PRINT
450  INPUT "WHICH PAGE TO LOAD (1
     /2) ? ";B$:A =  VAL (B$): PRINT

460  IF A <  > 1 AND A <  > 2 THEN
      PRINT : GOTO 450
465  PRINT : INPUT "INSERT CORREC
     T DISK AND <RTN> ";B$
470  PRINT  CHR$ (4)"BLOAD ";A$;"
     ,A$";(2000 * A)
480  GOTO 120
500  REM    ** USER WANTS TO GO *
     *
520  HOME : PRINT "END-": END
610  REM    ** WAIT A FEW SECONDS
     **
630  FOR Z2 = 1 TO 1000: NEXT Z2
640  RETURN
660  REM    **INSTRUCTTIONS **
680  HOME : INVERSE : HTAB 13: PRINT
     "HI-RES TOOLKIT ": NORMAL : PRINT

690  PRINT "HI-RES TOOLKIT IS A C
     OMBINED APPLESOFT AND MACHIN
     E LANGUAGE PROGRAM. TO USE I
     T, YOU JUST 'BRUN HI-RES TOO
     LKIT'. "
700  PRINT : PRINT "IF YOU WISH T
     O MODIFY THIS PROGRAM, TYPE
     'RUN 780' AFTER ALL THE CHAN
     GES HAVE BEEN MADE. THIS MAK
     ES CHANGES IN THE MACHINE LA
     NGUAGE ROUTINE SO THAT APPLE
     -SOFT WILL KNOW WHERE THE PR
     OGRAM IS. "
710  PRINT
720  PRINT "WITH HI-RES TOOLKIT,
     IT IS POSSIBLE TO (1) SWITCH
      THE HI-RES IMAGES BETWEEN P
     AGES (2) SUPERIMPOSE THE HI-
     RES IMAGES AND DISPLAY THEM
     ON PAGE 1 (3) INVERT THE IMA
     GES ON PAGE 1"
730  PRINT "(THE IMAGE CAN BE RES
     TORED BY INVERTING A SECOND
     TIME.)": PRINT
740  HTAB 17: PRINT "-ENJOY-": PRINT
     : INVERSE
750  HTAB 13: PRINT "<<PRESS SPAC
     E>> "; CHR$ (8);: GET A$: PRINT

760  RETURN
770  REM    **CHANGE ROUTINE**
780  TEXT : HOME : INVERSE : HTAB
     13
790  PRINT "CHANGE ROUTINE ": NORMAL
     : PRINT
800  PRINT "THIS IS TO BE USED AF
     TER ALL CHANGE  HAVE BEEN MA
     DE.": PRINT : PRINT : PRINT
     "<RTN> TO CONTINUE OR <ESC>
     TO ABORT:";: GET A$: IF A$ =
      CHR$ (27) THEN  PRINT : END

805  IF A$ <  > CHR$ (13) THEN  PRINT
     : VTAB 3: GOTO 800
810  PRINT : PRINT : PRINT "IF YO
     U WISH TO SAVE YOUR CHANGES,
      YOU MAY FORWARD ARROW OVER
     THIS LINE: ": PRINT : PRINT

820  POKE 2220, PEEK (175): POKE
     2230, PEEK (176)
840  PRINT " BSAVE HI-RES TOOLKIT
     ,A$800,L"( PEEK (175) +  PEEK
     (176) * 256) - 2048 + 2
850  VTAB 12: END
860  REM    ***END OF JOB ***
```

→

投稿規則

（一）本刊絕對公開，歡迎任何形式的投稿，包括：軟件程式，電腦硬件 REVIEW 及製作，商用、遊戲軟件 REVIEW，技巧使用，各種電腦語言介紹詳述等等。

（二）來稿以原稿紙單面橫寫爲合。

（三）軟件程式投稿，如程式有一定長度，請附同磁碟或磁帶一併寄來，程式必須加以清楚說明、用法及原理。

（四）寄來之磁碟或磁帶於稿件發表後即退回。

（五）稿件字數長短不限。

（六）請勿一稿兩投。

（七）所有投稿必須附上作者中英文眞實姓名、通信地址及電話（方便電腦紀錄），發表時筆名隨意。

（八）來稿可寄九龍中央郵局信箱 71193 號電腦時代收，信封面請註明〔投稿〕，或親身携來本刊。

如有疑問，請電3-312007 編輯部查詢

```
0800-  20 58 FC   JSR  $FC58
0803-  4C A7 08   JMP  $08A7
0806-  A9 20      LDA  #$20
0808-  A0 00      LDY  #$00
080A-  8D 23 08   STA  $0823
080D-  8D 36 08   STA  $0836
0810-  20 1F 08   JSR  $081F
0813-  C8         INY
0814-  C0 20      CPY  #$20
0816-  F0 06      BEQ  $081E
0818-  98         TYA
0819-  18         CLC
081A-  69 20      ADC  #$20
081C-  90 EC      BCC  $080A
081E-  60         RTS
081F-  A2 00      LDX  #$00
0821-  BD 00 3F   LDA  $3F00,X
0824-  48         PHA
0825-  AD 23 08   LDA  $0823
0828-  18         CLC
0829-  69 20      ADC  #$20
082B-  8D 33 08   STA  $0833
082E-  8D 3A 08   STA  $083A
0831-  BD 00 5F   LDA  $5F00,X
0834-  9D 00 3F   STA  $3F00,X
0837-  68         PLA
0838-  9D 00 5F   STA  $5F00,X
083B-  E8         INX
083C-  D0 E3      BNE  $0821
083E-  60         RTS
083F-  00         BRK
0840-  A9 20      LDA  #$20
0842-  A0 00      LDY  #$00
0844-  8D 5D 08   STA  $085D
0847-  8D 6D 08   STA  $086D
084A-  4C 76 08   JMP  $0876
084D-  C8         INY
084E-  C0 20      CPY  #$20
0850-  F0 06      BEQ  $0858
0852-  98         TYA
0853-  18         CLC
0854-  69 20      ADC  #$20
0856-  90 EC      BCC  $0844
0858-  60         RTS
0859-  A2 00      LDX  #$00
085B-  BD 00 3F   LDA  $3F00,X
085E-  48         PHA
085F-  AD 5D 08   LDA  $085D
0862-  18         CLC
0863-  69 20      ADC  #$20
0865-  8D 6A 08   STA  $086A
0868-  BD 00 5F   LDA  $5F00,X
086B-  1D 00 3F   ORA  $3F00,X
086E-  9D 00 3F   STA  $3F00,X
0871-  68         PLA
0872-  E8         INX
0873-  D0 E6      BNE  $085B
0875-  60         RTS
0876-  8D 70 08   STA  $0870
0879-  20 59 08   JSR  $0859
087C-  4C 4D 08   JMP  $084D
087F-  00         BRK
0880-  A9 20      LDA  #$20
0882-  A0 00      LDY  #$00
0884-  8D 9D 08   STA  $089D
0887-  8D A2 08   STA  $08A2
088A-  20 99 08   JSR  $0899
088D-  C8         INY
088E-  C0 20      CPY  #$20
0890-  F0 06      BEQ  $0898
0892-  98         TYA
0893-  18         CLC
0894-  69 20      ADC  #$20
0896-  90 EC      BCC  $0884
0898-  60         RTS
0899-  A2 00      LDX  #$00
089B-  BD 00 3F   LDA  $3F00,X
089E-  49 FF      EOR  #$FF
08A0-  9D 00 3F   STA  $3F00,X
08A3-  E8         INX
08A4-  D0 F5      BNE  $089B
08A6-  60         RTS
08A7-  A9 C4      LDA  #$C4
08A9-  85 67      STA  $67
08AB-  A9 27      LDA  #$27
08AD-  85 AF      STA  $AF
08AF-  85 69      STA  $69
08B1-  85 6B      STA  $6B
08B3-  85 6D      STA  $6D
08B5-  A9 12      LDA  #$12
08B7-  85 B0      STA  $B0
08B9-  85 6A      STA  $6A
08BB-  85 6C      STA  $6C
08BD-  85 6E      STA  $6E
08BF-  4C 66 D5   JMP  $D566
08C2-  00         BRK
08C3-  00         BRK
```

簡易快捷的移動記憶程式

你可曾試過用有效的方法把記憶搬移？這牽連到將文字檔案的資料由 DOS 緩衝器搬出搬入。雖然這個步驟可利用 BASIC 的 PEEK／POKE去做，但這個 PEEK／POKE 程式往往就要花上好幾分鐘才能完成僅長 8 K的資料。但假如利用 APPLE 監督程式內的程序去搬動 32 K BYTE 的資料却只需 2 秒多！實在是又快又容易。

這個監督程式的起步點在 $FE2C（65068，或—468）及一定要將Y暫存設置到零。辦法是用 APPLESOFT BASIC將監督程式的 SAVEY 網路 POKE 爲零，由此而啓動其中的副程式去還原指存器的值而跳至（JMP）移動的子程式。

下列之程式（程式一）是用來輸入緩衝區（PAGE 2）移動到高解像頁 PAGE 1 中。在 APPLE Ⅱ 的機械語言，以上的程式可被譯成 " 2000＜200.2FFM "。程式中的 A1 值是移動過程中的開始地址，而 A2 則是完結地址，A4 則是移動程式的目標地址值；而位址58，59則是用來存儲程式計數器的內存值；當暫存器的原來值被還原後，這個數據便被用在 JMP 副程式中。

```
10 A1 = 60: &  POKE A1,512: REM
     SET A1 ($3C/3C)=512 ($0200)
20 A2 = 62: &  POKE A2,767: REM
     SET A2 ($3E/3F)=767 ($02FF)
30 A4 = 66: &  POKE A4A4,8192: REM
   SET A4 ($24/43)=8192 ($2000)
40  POKE 71,0: REM  SAVEY ($47)=0($00)
50 PC = 58: &  POKE PC,65068: REM
     PC ($3A/3B)=65068 ($FE2C)
60  CALL  - 327: REM  CALL RESTOR
     E / JUMP ROUTINE
```

實際移動的副程式是在 $FE2C 這是一個很短的程式，它包括載入儲存操作，JSR 子程式至累加指梅（$FCB4）。程式本身只長 31 BYTE。

比較來說，程式一可寫成一個 BASIC 程式，不過這將會是一個較遲鈍的程式。因爲它用了 APPLE BASIC 之 & POKE 指令。這個程式能搬動任何記憶區的資料。進一步說，撇開了 & POKE 這個指令，一個更簡單的程式可以得到相同的效果，程式二。以下是程式中的執行指令。

& MOVE，SOURCE，TARGET，LENGTH。

其中 SOURCE 代表了開始位址，TARGET 代表要搬動的資料的開始地址，LENGTH 則代表要搬動資料的長度，而這長度則不能大過 32767 。

在未做移動之前，我們一定要查查監督程式的移動子程式以避免有錯誤的情形發生。在移動記憶其間監督程式不會視 A1 ，A2 或 A4 的值的。假如 A1 的值大過或等如A2的值，程式便會停止。同時，假如開始時 A1 經已大過 A2，則會被移走一個 BYTE 。要避免以上的情形發生，A4 的值應比原來要搬的值少一，這樣，這個移動的程序便能順利進行。

```
BLOAD REMOVE ROUTINE    程式（二）
*300.372

0300- A9 0B 8D F6 03 A9 03 8D
0308- F7 03 60 C9 4D D0 62 20
0310- B1 00 C9 4F D0 5B 20 B1
0318- 00 C9 56 D0 54 20 B1 00
0320- C9 45 D0 4D 20 B1 00 20
0328- BE DE 20 67 DD 20 52 E7
0330- 84 3C 85 3D 20 BE DE 20
0338- 67 DD 20 52 E7 84 42 85
0340- 43 20 BE DE 20 67 DD 20
0348- 52 E7 AA 30 21 A0 00 C6
0350- 50 A5 50 C9 FF D0 04 C6
0358- 51 30 12 B1 3C 91 42 E6
0360- 3C D0 02 E6 3D E6 42 D0
0368- E6 E6 43 D0 E2 60 4C 99
0370- E1 4C FF
```

監於這程式的操作沒有一系列的保護方法去監察，所以不是一個很妥善之方法，而程式二却能做到這個目的。

當執行程式二的時候（BRUN），$300—$30A是用來製定& 向量（AMPERSAND VECTOR）。$30B—$224 用作解譯 MOVE 指令，$327—$343用作解譯輸入資料（加開始／終止的位址和資料長度），同時製定記憶區的指標，這是爲了要防止記憶區被錯誤抹去；資料移動的長度是不能大過 32767（$7FFF）；如果資料長度等如零，則不會移動任何記憶資料。又，搬動資料目的地的開始地址是不能超過 $FFFF—$0000 否則MOVE 子程式便會終止執行，這是爲了保護零頁記憶區防止它被抹去。如果需要改變記憶零頁區的 BYTE ，搬動資料目的地的開始地址（TARGET ADDRESS）一定要是 $00 至 $FF 或 0 至 255。至於第 $34F—$36D ，則是用作配合監督程式的移動程式，雖然這會令整個程式二顯得較長，但其實它只用了31個 BYTE 。而整個程式二只需 1.6秒來移動 32767 BYTE 記憶。

基於以上的程式，高解像頁 PAGE 1 可以用以下之方法洗去：

「POKE 8192，0
& MOVE，8192，8193，8191」

如要將第二頁記憶移動到第一頁則是：

& MOVE，16384，8192，8191

如要將第一頁搬去第二頁，則是：

& MOVE，8192，16384，8192

至此，相信讀者們可以很方便地將記憶區移動，但要留心避開 I／O頁（$C000）；同時要留意記憶區地址 2048到38400，避免摧毀 DOS 和記憶零頁區 PAGE 0

程式三是一個示範程式，示範如何搬移記憶往任何位置。

簡易快捷的移動記憶程式

```
 1 REM **************************
 2 REM *          程式(三)        *
 3 REM *       REMOVE MEMORY      *
 5 REM *                          *
 6 REM **************************
10  HOME : PRINT "LOADING & MOVE
    ROUTINE TO PAGE 3"
20  GOTO 40
30 H$ = H$ + "ND9C6G": FOR I = 1 TO
    LEN (H$): POKE 511 + I, ASC
    ( MID$ (H$,I,1)) + 128: NEXT
    : POKE 72,0: CALL  - 144: RETURN

40 H$ = "0300:A9 0B 8D F6 03 A9 0
    3 8D F7 03 60 C9 4D D0 62 20
     B1 00 C9 4F D0 5B 20 B1 00
    C9 56 D0 54 20 B1 00 C9 45 D
    0 4D 20 B1 00 20 BE DE 20 67
     DD 20 52 E7 84 3C 85 3D 20
    BE DE 20 67 DD 20 52 E7 84 4
    2 85 ": GOSUB 30
50 H$ = "0340:43 20 BE DE 20 67 D
    D 20 52 E7 AA 30 21 A0 00 C6
     50 A5 50 C9 FF D0 04 C6 51
    30 12 B1 3C 91 42 E6 3C D0 0
    2 E6 3D E6 42 D0 E6 E6 43 D0
     E2 60 4C 99 E1 4C C9 DE": GOSUB
    30
60  POKE 1014,11: POKE 1015,3
70 P1 = 8192:P2 = 16384:A = 128:B
    = 40:L = 8192: HOME
80  POKE P1,0: & MOVE,P1,P1 + 1,L
    - 1: & MOVE,P1,P2,L
90  HGR : POKE P1,42: & MOVE,P1,P
    1 + 1,L - 1: FOR J = 1 TO 50
    0: NEXT
100 POKE P1,85: & MOVE,P1,P1 + 1
    ,L - 1: FOR J = 1 TO 500: NEXT

110 POKE P1,42: POKE P1 + 1,85: &
    MOVE,P1,P1 + 2,L - 2: FOR J =
    1 TO 500: NEXT
120 POKE P1,85: POKE P1 + 1,42: &
    MOVE,P1,P1 + 2,L - 2: FOR J =
    1 TO 500: NEXT
130 HGR2 : POKE P2,170: POKE P2 +
    1,213: & MOVE,P2,P2 + 2,B -
    2
140 FOR J = 0 TO 2:X = P2 + J *
    B: FOR N = 0 TO 7: FOR I = N
     TO N + 56 STEP 8
150 & MOVE,P2,X + I * A,B: NEXT
    I,N,J
160 FOR I = 1 TO 5:X =  PEEK (49
    236): FOR J = 1 TO 500: NEXT
    :X =  PEEK (49237): FOR J =
    1 TO 500: NEXT : NEXT
170 HOME : TEXT : HTAB 10: PRINT
    "I LOVE THIS COLOR ": HGR
180 POKE P2,213: POKE P2 + 1,170
    : & MOVE,P2,P2 + 2,B - 2
190 FOR J = 2 TO 0 STEP  - 1:X =
    P1 + J * B: FOR N = 7 TO 0 STEP
     - 1: FOR I = N + 56 TO N STEP
     - 8
200 & MOVE,P2,X + I * A,B: NEXT
    I,N,J
```

□

展覽會採用「輕觸電腦」入場登記

一個名的「香港社會服務鱗爪展覽」將於七月廿一日至廿三日於香港大會堂舉行，展覽會得到十多個社會服務團體參與，介紹社會服務概念及其機構之工作。

展覽會的其中一項特色是大會採用最新的「輕觸電腦」作入場統計，市民可親自輕觸電腦螢幕便可登記入場人次。此外，更利用先進電腦系統製作一個完備的「香港社會服務」電腦資料庫。參觀者只需要等候片刻便可獲得資料，資料包括機構所提供的服務，工作區域等等。

會中更每日放映有關社會問題的電影、幻燈及錄影帶，內容廣泛包括青少年問題、幼兒護理、心理衞生、輔導服務等等。凡此種種，都是希望透過不同的表達方式，爲市民提供利用社會服務之途徑。

是次展覽整個構思着重專重各個參展社團服務特色爲根基，各社會服務團體共抱一個社會異象，而以不同角色去參與及改善社會問題，從而達到「造福人群、服務社會」之同一目標。

香港社會服務鱗爪展覽將在七月廿一日（星期六）至二十三日（星期一）假香港大會堂展覽廳舉行，星期六下午三時開幕，星期日及一則在上午十時開放。是由香港教育促進中心主辦。□

以GRAPHICS MAGICIAN編寫的遊戲：DEPTH CHARGE

麥法基

編者註：每位蘋菓電腦用家相信都會搜集不少的遊戲軟件程式，但是各位有沒有想過自己去編寫一些具有職業水準的ARCADE GAME呢？本文作者麥法基先生非常熱衷於電腦，並希望透過個人的經驗把一些編寫遊戲軟件的心得技巧公開給大家。麥先生是利用GRAPHICS MAGICIAN作爲輔助來創造程式中的動畫部份。

由於本欄只著重於介紹遊戲編寫的技巧，故此我們不打算將DEPTH CHARGE遊戲的程式列表刊出，讀者們可在同期的程式磁碟中找到這個遊戲。

假如讀者對本欄有疑問，歡迎來信轉交麥先生解答。

筆者在上一期曾經介紹一個賽車遊戲，在編寫程式方面結構較爲簡單，不過已具備有一般編寫ARCADE GAME的技巧了。無可否認在大多數市售的遊戲中，一個好的ANIMATION PACKAGE十分重要。你所熟悉的THRESHOLD，CROSS FIRE，ARDY，SPACE，QUARK等都需要ANIMATION PACKAGE來創造精彩的動畫技巧。

從這一期開始，筆者介紹另一個名爲DEPTH CHARGE的遊戲，這個遊戲模倣一般電子遊戲中心其中一個以深水炸彈攻擊潛水艇的遊戲。玩法十分簡單並不需要在這裡加以說明，不過由於程式比較長及複雜，所以程式的解釋需要分期數刊登。有興趣研究遊戲的朋友，可以購買電腦時代程式磁碟，這樣可以省去鍵入程式的時間，事實上，鍵入太長的機械語程式是件十分麻煩而沈悶的工作。

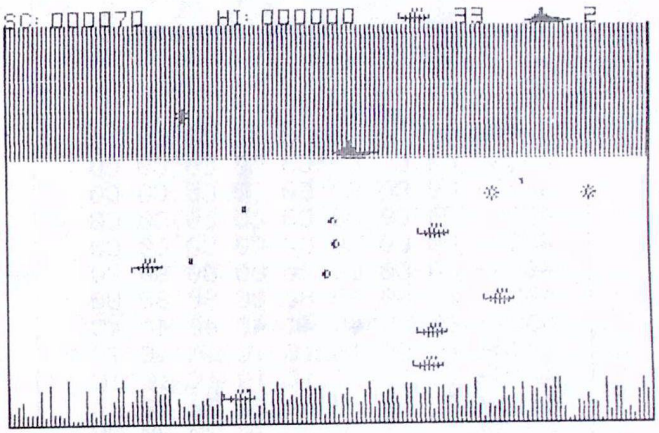
整個程式仍然包括三部份，BASIC，機械語言及GRAPHICS MAGICIAN的動畫程式。這一期首先研究如何編製動畫程式部份，因爲倘若對動畫程式不太明白的話，對編寫整個程式會有某些程度上的困難。原則上，動畫程式以一種十分簡單的方法把每一個圖形繪畫在螢幕上，在未進行繪畫工作之前，它會到每個指定的記憶位址找尋資料，例如圖形、路綫、碰撞、消圖等，如此整個程式加以迴路便構成基本的遊戲骨幹了。當然，在每一動不同的遊戲當中，你可能有數個至數十個不同的物體在螢幕上移動，一般BASIC語言根本無法達成完善的工作，這樣就非倚靠機械語言不可了。

DEPTH CHARGE的遊戲是這樣的，在深海下面有很多快慢不同的潛艇向海面發射魚雷及飛彈，而你控制的一艘戰艦便是它攻擊的對象，當然你亦不斷的放射深水炸彈向他們反擊，不過要注意的是潛艦會有時發射一些在海面會回轉攻擊的導向飛彈，稍不留神便會被擊沉，而海底亦會不斷放出水雷同時向你攻擊，這樣遊戲便會十分緊張刺激了。整個遊戲有超過二十個LEVEL，如何應付便靠你靈活的反應了。

編製動畫程序

至於怎樣編製動畫程式呢？參考FIG#1至FIG#31；總共有三十一個圖形，其中包括戰艦、潛艇、魚雷、飛彈、爆炸及字形等。這個遊戲需要整個高解像畫面；事實上，它連第一頁和第二頁也全部用上了，這樣我們再不能應用到TEXT的文字幕，所以連一個問號的圖形都須要加以編製出來。

遊戲繪圖以BLOCK WITH BACKGROUND形式



編製，所以圖形可需要在上下左右加上邊緣輪廓，如此圖形移動時便毋須再造消圖的工作，這樣的動畫可減少時間而加快了很多速度，更加除去閃動的現象。在這裡筆者不再詳述如何利用SHAPE EDITOR，因爲GRAPHICS MAGICIAN手冊內已有說明。要留意的是FIGURE#2的潛艇圖形，單數圖形用上面那一個，而雙數圖形用下面一個，目的是使潛艇移動時船漿有搖動效果。完成編製圖形後記著把圖形表分別存入磁碟內，圖形表的FILE NAME在FIGURE#1至#31內有列出。

第二部份便是PATH TABLE，由於GRAPHICS MAGICIAN的PATH EDITOR比較難於在文字上詳細說明，筆者把已經造好的PATH TABLE在這裡列出，讀者只須把路綫表內容鍵入然後BSAVE到磁碟上即可，留意的是FILE NAME必須有PTH字樣。PATH TABLE比較短，相信鍵入不會有困難。

留意LOAD入次序

最後是編製ANIMATION程式了。上期筆者曾經提及把各種TABLE LOAD入ANIMATION EDITOR的次序十分重要，因爲該EDITOR是根據你LOAD

DEPTH CHARGE

SHIP PATH.PTH,A$4000,L$2

```
4000- 08 00
```

TOPPEDO PATH.PTH,A$4000,L$2

```
4000- 08 00
```

SUBMARINE PATH.PTH,A$4000,L$2

```
4000- 08 00
```

XBOMB PATH1.PTH,A$4000,L$B

```
4000- 82 82 82 82 82 82 88 88
4008- 88 88 00
```

RBOMB PATH1.PTH,A$4000,L$61

```
4000- C0 C0 C0 C0 C0 C0 C0 C0
4008- C0 C0 C0 C0 C0 C0 C0 C0
4010- C0 C0 C0 C0 C0 C0 C0 C0
4018- C0 C0 C0 C0 C0 C0 C0 C0
4020- C0 C0 C0 C0 C0 C0 C0 C0
4028- C0 C0 C0 80 80 80 80 80
4030- 80 80 80 80 80 80 80 80
4038- 80 4C 4C 4C 4C 4C 4C 4C
4040- 4C 4C 1C 1C 1C 1C 1C 1C
4048- 1C 1C 1C 1C 1C 1C 1C 1C
4050- 20 20 20 20 20 20 20 20
4058- 20 20 20 20 20 20 20 20
4060- 00
```

XBOMB PATH2.PTH,A$4000,L$F

```
4000- 88 88 88 88 88 88 82 82
4008- 82 82 82 82 82 82 00
```

XBOMB PATH3.PTH,A$4000,L$B

```
4000- 82 82 82 82 82 82 88 88
4008- 88 88 00
```

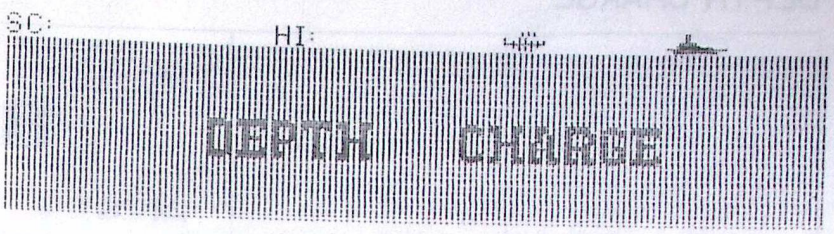
RBOMB PATH2.PTH,A$4000,L$63

```
4000- C0 C0 C0 C0 C0 C0 C0 C0
4008- C0 C0 C0 C0 C0 C0 C0 C0
4010- C0 C0 C0 C0 C0 C0 C0 C0
4018- C0 C0 C0 C0 C0 C0 C0 C0
4020- C0 C0 C0 C0 C0 C0 C0 C0
4028- C0 C0 C0 C0 80 80 80 80
4030- 80 80 80 80 80 80 80 80
4038- 43 43 43 43 43 43 43 43
4040- 43 43 13 13 13 13 13 13
4048- 13 13 13 13 13 13 13 13
4050- 13 13 13 13 20 20 20 20
4058- 20 20 20 20 20 20 20 20
4060- 20 20 00
```

CHARGE PATH.PTH,A$4000,L$D

```
4000- 28 28 28 28 28 28 22 22
4008- 22 22 22 22 00
```

DEPTH CHARGE



入的次序設定圖形及路綫號碼，稍有錯誤會導致程式不能運行。現詳列次序如下：

（一）BOOT 起 GRAPHICS MAGICIAN按（A）進入 ANIMATION EDITOR。

（二）按（2）選擇 BLOCK WITH BACKGROU—ND 繪圖法，換上自己的磁碟。

（三）按（L）及（S）鍵 LOAD 入圖形表，首先把 FIGURE #1 至 FIGURE #6 依次序 LOAD 入，因爲這些圖形均需要跟路綫表移動。

（四）完成第三步驟後再按（L）及（P）鍵 LOAD 入路綫表 （ PATH ）。首先是 SHIP PATH 路綫表，然後把 SUBMARINE PATH 分6次 LOAD 入（因爲有6艘潛艇在螢幕上移動），跟着是兩條 RBOMB PA—TH ，再把 TOPPEDO PATH 分5次 LOAD 入（因爲有五枝魚雷同時發射），最後把其餘的 XBOMB PAT—H及 CHARGE PATH 依次序 LOAD 入。

（五）按（L）及（S）鍵再把其餘 FIGURE #7 至 FIGURE #31的圖形表 LOAD 入。

（上述第四項步驟因爲 GRAPHICS MAGICIAN 會自動把 FILE NAME 後面的 PTH 字樣加入，所以 LOAD 入時不必再打 PTH ）。

ANIMATION EDITOR ENTRIES TABLE

OBJ	SHAPE	X,Y-CORD		PATH	START	LOCATION
0	0	140	63	0	255	0	0
1	1	300	100	1	255	0	0
2	1	400	115	2	255	0	0
3	1	500	130	3	255	0	0
4	1	600	175	4	255	0	0
5	1	700	160	5	255	0	0
6	1	800	145	6	255	0	0
7	4	200	191	7	254	0	0
8	4	20	191	8	254	0	0
9	2	10	190	9	255	0	0
10	2	30	190	10	255	0	0
11	2	60	190	11	255	0	0
12	2	70	190	12	255	0	0
13	2	90	190	13	255	0	0
14	5	22	190	14	255	0	0
15	5	100	190	15	255	0	0
16	5	200	190	16	255	0	0
17	3	70	70	17	255	0	0
18	3	80	70	18	255	0	0
19	3	130	70	19	255	0	0

輸入資料

相信現在你已經把以上程序做妥了，就讓我們把資料再供應給 ANIMATION EDITOR吧。按（N）鍵後 APPLE 會要求輸入 NEW OBJECT 的資料，請把 AN—IMATION EDITOR ENTRIES TABLE的資料依次序輸入，直至有 20 個 OBJECT爲止。如此整個 ANI—MATION ROUTINE經已大功告成了，當然不要忘記把 ROUTINE按（S）鍵以 DEPTH CHARGE 名字 SAVE 到磁碟上。

整個 ROUTINE 佔用由 31573至 38399（$7B55至 $95FF）的記憶，超過2 K所以十分長。不過用上述方法編製便不用鍵入這樣長的 MACHINE CODE ，於是我們便可集中精神去創造遊戲的其他部份。當然，單靠這個 ROUTINE 我們不能立即完成遊戲，現在就研究一下究竟 ROUTINE 包括些甚麼秘密吧。

首先鍵入 LISTING #1 的 BASIC 程式，然後運行。假如 ROUTINE 沒有錯的話，你會發現一些不斷變化的數字在螢幕上向左移動。原因是我們改變了 ROUTINE 的OBJECT LIST ，位置由 37888 開始（ $9400至 $9420）。每一個位址包括一個圖形號碼，這些號碼在我們編製動畫程式時已經預先輸入了，而每一個位址亦代表一個物體，所以我們可以總共有 32 個物體在螢幕上同時出現。這些位址的值（ VALUE）由圖形號碼所決定，倘若值是 255（$FF）的話，表示 OBJECT LIST 完結，動畫程式便不再繼續從 LIST 內尋找圖形。換言之，螢幕上物體數目可以因改變這些值而有所不同。看看 LISTING #1，行號 15 我們把255的值放入 37889（$9401）的位址內，所以螢幕便只有一個物體移動。行號 20 我們把值暫設爲9，然後由行號 50 再把值增加至 18 ，參看 FIGURE #10 的圖形號碼是9，如此螢幕便顯示由0至9的圖形不斷向左移。（由於 37889 的值是 255，所以動畫程式只取 37888 的值。螢幕便只能有一個物體存在）

雖然我們很容易在程式運行中改變圖形，不過有些少限制，就是如果以 BLOCK 的繪圖法，圖形大少必須一樣，否則會存在有較大圖形的殘餘影像。

PATH的示範

現在鍵入 LISTING #2來看看如何應用 PATH 的示範。運行此程式你會看見螢幕上的物體從下面向上行一次然後向左右方反彈移動，參看程式行號 24 我們把圖形改爲4的飛彈圖形。行號 25 我們把 PATH LOCATION（$9440至$949F是每一物體的路綫指針）改爲指向第14條路綫表的起始位置。每一條路綫都會由 ANIMATION EDITOR 在 LOAD 時放在 APPLE 的某一記憶位置上。不過我們在那裡可以找到這些位置呢？在 $9500至 $95

DEPTH CHARGE

FIGURE #1 - SHIP
SHAPE NO.#0 X=26 Y=8

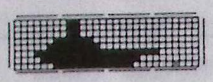
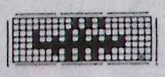


FIGURE #2 - SUBMARINE
SHAPE NO.#1 X=20 Y=7



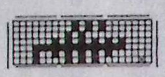


FIGURE #3 - TOPPEDO
SHAPE NO.#2 X=8 Y=9

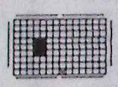


FIGURE #4 - CHARGE
SHAPE NO.#3 X=8 Y=8

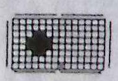


FIGURE #5 - GUIDED TOPPEDO
SHAPE NO.#4 X=13 Y=13

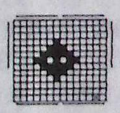


FIGURE #6 - BOMB
SHAPE NO.#5 X=11 Y=11

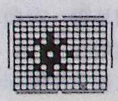


FIGURE #7 - EXPLOSION
SHAPE NO.#6 X=9 Y=7

FIGURE #8 - SHEXPLOSION
SHAPE NO.#7 X=14 Y=9

FIGURE #9 - EXPLOSION2
SHAPE NO.#8 X=12 Y=6

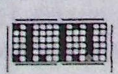


FIGURE #10 - ZERO
SHAPE NO.#9 X=5 Y=7

FIGURE #11 - ONE
SHAPE NO.#10 X=5 Y=7

FIGURE #12 - TWO
SHAPE NO.#11 X=5 Y=7

FIGURE #13 - THREE
SHAPE NO.#12 X=5 Y=7

FIGURE #14 - FOUR
SHAPE NO.#13 X=5 Y=7

FIGURE #15 - FIVE
SHAPE NO.#14 X=5 Y=7

FIGURE #16 - SIX
SHAPE NO.#15 X=5 Y=7

?SYNTAX ERROR

FIGURE #17 - SEVEN
SHAPE NO.#16 X=5 Y=7

FIGURE #18 - EIGHT
SHAPE NO.#17 X=5 Y=7

FIGURE #19 - NINE
SHAPE NO.#18 X=5 Y=7

FIGURE #20 - A
SHAPE NO.#19 X=7 Y=7

FIGURE #21 - D
SHAPE NO.#20 X=7 Y=7

FIGURE #22 - E
SHAPE NO.#21 X=7 Y=7

FIGURE #23 - G
SHAPE NO.#22 X=7 Y=7

FIGURE #24 - M
SHAPE NO.#23 X=7 Y=7

FIGURE #25 - O
SHAPE NO.#24 X=7 Y=7

FIGURE #26 - R
SHAPE NO.#25 X=7 Y=7

FIGURE #27 - S
SHAPE NO.#26 X=7 Y=7

FIGURE #28 - T
SHAPE NO.#27 X=7 Y=7

FIGURE #29 - V
SHAPE NO.#28 X=7 Y=7

FIGURE #30 - Y
SHAPE NO.#29 X=7 Y=7

FIGURE #31 - QUESTION
SHAPE NO.#30 X=7 Y=7

LISTING #1 SHAPE DEMO

```
10  HGR2 : HGR : PRINT  CHR$ (4);
    "BLOAD DEPTH CHARGE.ANM"
15  POKE 37889,255
20 N = 9
30  POKE 37888,N
40  CALL 36928
50 N = N + 1: IF N > 18 THEN N =
    9
60  FOR I = 0 TO 100: NEXT
70  GOTO 30
```

LISTING #2 PATH DEMO

```
10  HGR2 : HGR : PRINT  CHR$ (4);
    "BLOAD DEPTH CHARGE.ANM"
20  POKE 37889,255
24  POKE 37888,4
25  POKE 37952,0: POKE 37953,175:
    POKE 37954,136
30 X =  PEEK (38048)
40  IF X < 2 THEN  POKE 35220,2
50  IF X > 36 THEN  POKE 35220,8
60  CALL 36928
70  FOR I = 0 TO 50: NEXT
80  GOTO 30
```

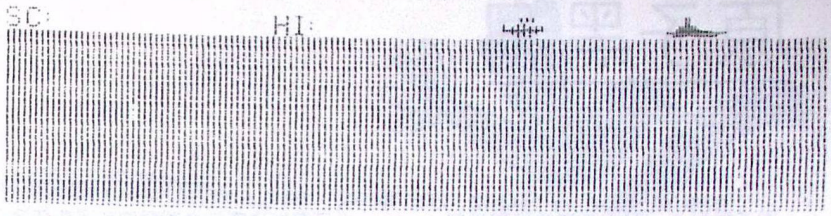
LISTING #3 PATH ADDRESS FINDER

```
10  TEXT : HOME : FLASH : PRINT "
    HAVE YOU BLOADED YOUR ANIMAT
    ION ROUTINE (Y/N)"
20  GET A$: IF A$ <  > "Y" THEN
    NORMAL : END
25  NORMAL : HOME
30  INPUT "OBJECT NUMBER?";N
40  INPUT "HOW MANY PATHS?";M
50  IF M < 0 OR M > 3 THEN 40
60 P1 =  PEEK (38144 + 7 * N):P2 =
    PEEK (38145 + 7 * N):P3 = P
    1 * 256 + P2
70 Q1 =  PEEK (38146 + 7 * N):Q2 =
    PEEK (38147 + 7 * N):Q3 = Q
    1 * 256 + Q2
80 R1 =  PEEK (38148 + 7 * N):R2 =
    PEEK (38149 + 7 * N):R3 = R
    1 * 256 + R2
90  IF M = 1 THEN  GOSUB 200: GOTO
    30
100  IF M = 2 THEN  GOSUB 200: GOSUB
    210: GOTO 30
110  IF M = 3 THEN  GOSUB 200: GOSUB
    210: GOSUB 220
120  GOTO 30
200  PRINT "PATH NO.1";" ";"HI=";
    P1;" ";"LO=";P2;" ";"ADDRESS
    =";P3: RETURN
210  PRINT "PATH NO.2";" ";"HI=";
    Q1;" ";"LO=";Q2;" ";"ADDRESS
    =";Q3: RETURN
220  PRINT "PATH NO.3";" ";"HI=";
    R1;" ";"LO=";R2;" ";"ADDRESS
    =";R3: RETURN
```

LISTING #4 PATH DEMO1

```
10  HGR2 : HGR : PRINT  CHR$ (4);
    "BLOAD DEPTH CHARGE.ANM"
20  POKE 37889,255
24  POKE 37888,4
25  POKE 37952,0: POKE 37953,175:
    POKE 37954,136
27 C = 0
30 X =  PEEK (38048)
35 C = C + 1: PRINT C: IF C > 100
    THEN  POKE 37952,254
40  IF X < 2 THEN  POKE 35220,2
50  IF X > 36 THEN  POKE 35220,8
60  CALL 36928
70  FOR I = 0 TO 50: NEXT
80  GOTO 30
```

DEPTH CHARGE



DF便可以找到了。在這裡每一個物體佔用位址的7個位元組(BYTE)，爲首的6個位元組顯示 PATH 的記憶位置以高低位元指示。(每一條記憶位址可分爲高位元和低位元，例如$8704的高位元是$87，低位元是$04)。最後的位元組是旗號，決定該物體的路綫是否繼續，消圖還是固定。假如物體只須一條路綫的話，則第三位元組便順理成章改爲旗號處理了。雖然我們知道在$9500可以找到路綫位置，總不能亂摸亂撞去尋找吧！你可以利用 LISTING $3 的 PATH ADDRESS去找出每個物體的路綫起始位置，對於以後應用 GRAPHICS MAGICIAN 十分有用。(記着首先把任何已經造好的ANIMATION ROUTINE存放入APPLE才可以運用。)現在再看 LISTING#2 的行號 25，我們已經知道第14條路綫(即第14號物體的路綫)的H1位元是135，而LO位元是4，於是我們設定這條路綫的號碼爲0，這就是行號25的來源了。每一個物體的路綫位置除了在$9500所列出之外，$9440至$949F亦是每一物體路綫的行走位址。$9500的 PATH LIST只是供應路綫的位置資料而已，而$9440的 PATH LOCATION 是物體在移動時的位址。當物體移動一步時，它的位址便會有所改變，因爲路綫表可能包括一個至幾百個以上的移動單位，所以假使你的路綫表越長(即移動步驟越多)則所須的記憶會越多。下面再詳述關於移動單位(MOVE)的問題。

工作原理

PATH LOCATION 的工作情形是這樣的。當動畫程式開始運行時，每一物體的移動由 PATH LOCATION決定，所以它的指針會指向第一條路綫及路綫的位址，位址資料以LO—H1方式存於第二及第三位元組，每個物體順序佔用三個位元組，第一位元組是路綫號碼(存放在PATH LIST的號碼)。路綫完畢後再從PATH LIST找尋另外一條路綫表的起始位址去繼續移動，倘若沒有其他路綫表跟隨，則從旗號去決定是否循環或消圖等。很明顯地由於 PATH LOCATION和路綫表內的移動單位有關，我們可以把它的指針指向路綫表內任何一個移動單位來改變物體的移動路綫。換言之，指針並非一定是路綫表開始的位址，你可以使它指向第二個，第三個或任何一個步驟。例如 SPACE QUARK 這個遊戲的怪獸移動時每隻怪獸都用同一條路綫，但起步點不同，所以你會看見怪獸一個跟一個在螢幕上排隊移動。

在這裡有一個有趣的問題，假如你設計一個打怪獸的遊戲，需要怪獸從螢幕以外進入螢幕內循環打圈移動，但不容許怪獸再次跑出螢幕外面，應該如何利用 PATH LOCATION來達成這個效果呢？下一期會給你一個答案。

再談談有關 PATH LOCATION，在LISTING#2內加入行號27及第35而刪去行號70變成 LISTING#4 的 PATH DEMO I。運行後發覺當C的值大於100時，物體忽然靜止不動。原因是在行號35我們把254的值存入 PATH LOCATION 第一個物體的第一個位元組內，倘若這個位元組的值不是路綫號碼而是254的話則 GRAPHICS MAGICIAN 會把它認爲是物體靜止旗號而跳過繪圖工作，如此使我們更有效地運用動畫程式。

現在請刪去行號27及35，然後研究行號30。由38048($94A0至$94FF)開始是每個物體的水平及垂直座標位置資料。動畫程式開始着手繪圖之前，首先在這個位址找尋物體在螢幕上的位置。隨着物體位置的改變，這些位址的值亦跟着改變，使我們可以把這些值作爲各種測試的運算。

38048 是 OBJECT LOCATION 位址表。每個物體佔用三個位元組，第一位元組是該物體水平位置的位元組(BYTE)值，第二個是水平位置位元(BIT)值，而第三個是垂直位置值。(讀者想清楚知道關於水平座標的位元組與位元關係，請參閱施銘威的微電腦徹底研究第二章第五冊)由於位元組值在螢幕只能顯示由0至39的值，倘若大於39的值會使物體在螢幕以外移動。LISTING $2行號30便是測試該位元組的值，而行號40和50是測試它在水平位置的界限。但爲甚麼我們要在34793的位址存入兩個不同的數值呢？這就是我們要研究的移動單位問題了。

PATH的營造

首先讓我們觀察一下究竟 PATH 是如何形成的。每一條 PATH 包括一個至數百個移動單位(MOVE)，每一個 PATH 內的移動單位以每一個位元組(BYTE)的值來決定它的速度和方向。參看 SHIP PATH，PTH那一條路綫，第一個位元組值是08，而第二個是00。它表示這條路綫只有一個移動單位，然後00則表示路綫完畢。當路綫完畢後就由 PATH LIST 的旗號決定繼續把路綫循環，現在請看移動單位的值如何決定速度和方向。

由於每個移動單位的位元組包括8個位元(BIT)，所以 GRAPHICS MAGICIAN 把8個位元分成4組，每組包括一對位元，而4組位元表示四個不同方向的移動。由此可見既然每組位元只有兩BIT，所以只能表示0至3的數值，這亦是物體可以移動0至3個位元的距離(二進制的01是十進制的1，10是十進制的2，11則是十進制的3，相信大家對二進制數目必定有所認識吧)

試看08這個數值如何表示移動情形：

$08=00001000(二進制)

(下轉P.50)

把以上數值分爲四組位元如下：

原子黑洞

原子黑洞是一個需要高度智慧推理頭腦的遊戲，玩者必須憑發生的各種反應來追尋黑洞內原子的所在位置。

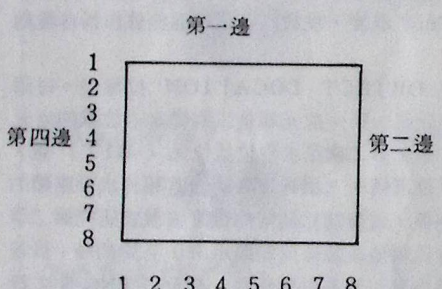
原子黑洞是由一個人行乘八行的座標組成，遊戲要求玩者利用發射原子來進行撞擊存在於黑洞內的原子，這些原子是看不到的，但這些原子相遇時，玩者所射出的原子會產生下列三項其中之一的反應：

（1）被吸納而消失

（2）折射向別個方向

（3）反彈回頭

玩者可選擇四邊任何一邊來發射，發射位置有八個。（參看圖1）

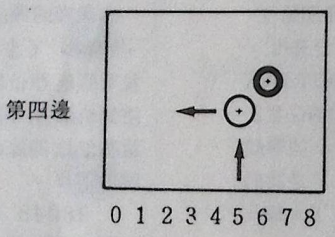


圖（1）

所發射之原子會被吸納的可能情況是將所發射的原子前進方向途中有看不見的原子。假如中途沒有任何原子，則會直貫穿過對面那一邊。

第二種情形是折向另一方向前進。假如在所發射的原子對角綫右手邊早有原子存在，則玩者發射的原子會折射向左手邊，如圖2所示。

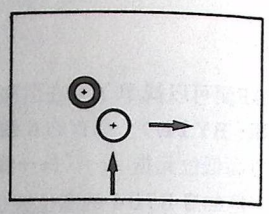
圖（2）



與之相反，當發射之原子對角綫左手邊苦有原子存在，則這枚原子（玩家原子）則會被折射到右向方向。圖（3）清楚顯示上述所說。

假如在洞內的原子是靠近四邊牆任何一邊，則在其隔鄰所發射的原子會被反彈回發射的位置。

圖（3）



若黑洞內原子的數目越多，則所產生的反射結果會是有趣而複雜。

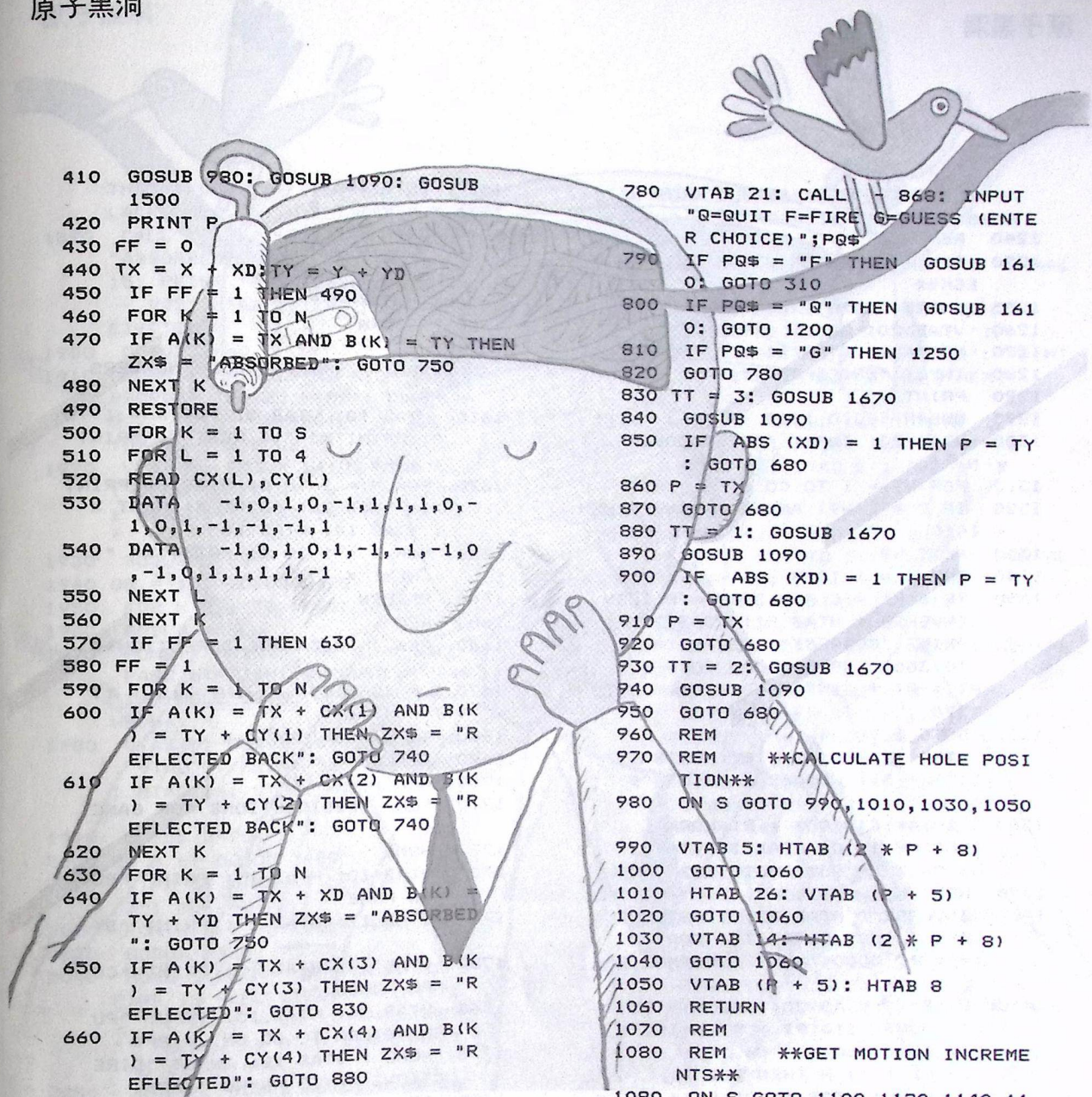
黑洞內看不見之原子數目是於遊戲開始時，由玩者決定。

原子黑洞遊戲的計分方法如下：

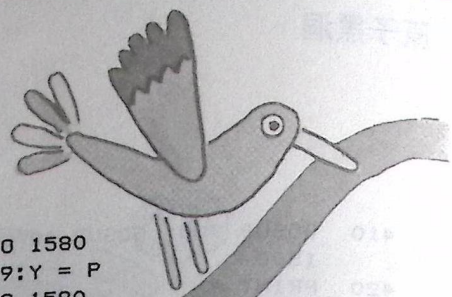
	1至4枚原子	5至8個原子
發射原子	減20分	減10分
錯誤猜估	減100分	減200分
正確猜估	加500分	加1000分

本程式在程式磁碟中的FILENAME是BLACK HOLE □

```
10  REM   **********************
20  REM   *      BLACK HOLE     *
40  REM   * COPYRIGHT (C) 1984 *
50  REM   * BY COMPUTING - AGE *
60  REM   **********************
80  GOSUB 1720
90  DEF  FN R(Q) =  INT (8 *  RND
    (3)) + 1
100  HOME : VTAB 12: ONERR  GOTO
    100
110  INPUT "ENTER # OF ATOMS (UP
    TO 8) ";N
120  IF N <  = 0 OR N > 8 THEN  VTAB
    4: PRINT "TRY AGAIN": CALL  -
    958: VTAB 12: GOTO 110
130 DI =  INT ((N + 3) / 4)
140  FOR K = 1 TO 8:A(K) = 0:B(K)
     = 0:: NEXT
150 A(1) =  FN R(3):B(1) =  FN R(
    3)
160  FOR K = 2 TO N
170 CX =  FN R(3):CY =  FN R(3)
180  FOR L = 1 TO K - 1
190  IF A(L) = CX AND B(L) = CY THEN
    170
200  NEXT L
210 A(K) = CX:B(K) = CY
220  NEXT K
230  HOME
240  REM
250  REM    **DRAW THE BLACK HOLE
    **
260  GOSUB 1610
270  VTAB 2: HTAB 15: PRINT "SIDE
     1"
280  VTAB 9: PRINT "SIDE 4"
290  VTAB 9: HTAB 29: PRINT "SIDE
     2"
300  VTAB 17: HTAB 15: PRINT "SID
    E 3"
310  VTAB 18: CALL  - 958: HTAB 1

320  HTAB 1: VTAB 19: PRINT "POIN
    TS = "PT
330  REM
340  REM    **ENTER FIRING POSITI
    ON**
345  ONERR  GOTO 350
350  VTAB 21: INPUT "ENTER SIDE #
     ";S
360  IF S > 4 OR S < 0 THEN 310
370 PT = PT -  INT (20 / DI)
375  ONERR  GOTO 380
380  VTAB 21: CALL  - 958: INPUT
    "ENTER POSITION NUMBER ON TH
    AT SIDE ";P
390  IF P > 8 OR P < 0 THEN 380
400  INVERSE
```

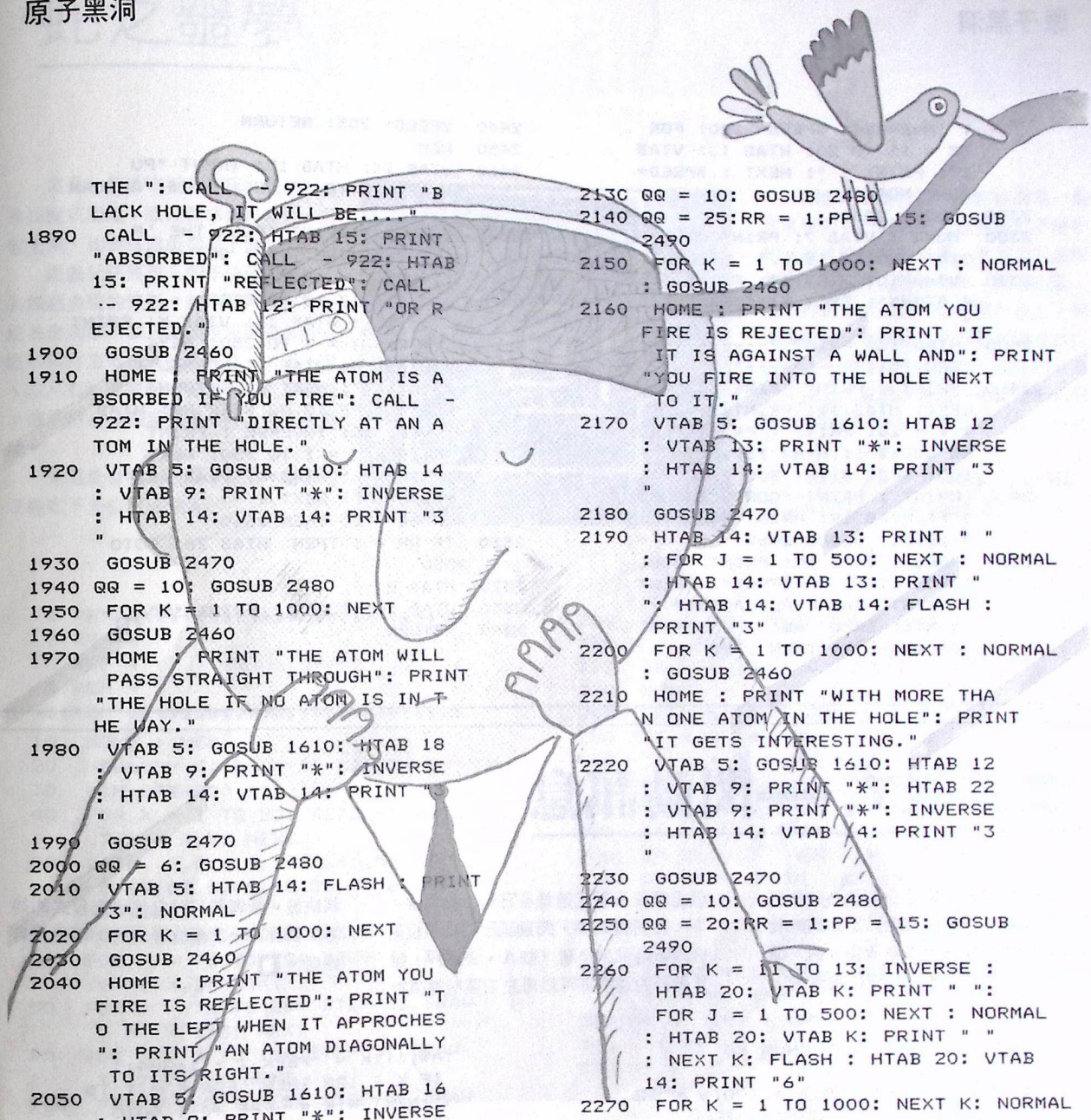


```
410  GOSUB 980: GOSUB 1090: GOSUB
     1500
420  PRINT P
430 FF = 0
440 TX = X + XD:TY = Y + YD
450  IF FF = 1 THEN 490
460  FOR K = 1 TO N
470  IF A(K) = TX AND B(K) = TY THEN
     ZX$ = "ABSORBED": GOTO 750
480  NEXT K
490  RESTORE
500  FOR K = 1 TO S
510  FOR L = 1 TO 4
520  READ CX(L),CY(L)
530  DATA   -1,0,1,0,-1,1,1,1,0,-
     1,0,1,-1,-1,-1,1
540  DATA   -1,0,1,0,1,-1,-1,-1,0
     ,-1,0,1,1,1,1,-1
550  NEXT L
560  NEXT K
570  IF FF = 1 THEN 630
580 FF = 1
590  FOR K = 1 TO N
600  IF A(K) = TX + CX(1) AND B(K
     ) = TY + CY(1) THEN ZX$ = "R
     EFLECTED BACK": GOTO 740
610  IF A(K) = TX + CX(2) AND B(K
     ) = TY + CY(2) THEN ZX$ = "R
     EFLECTED BACK": GOTO 740
620  NEXT K
630  FOR K = 1 TO N
640  IF A(K) = TX + XD AND B(K) =
     TY + YD THEN ZX$ = "ABSORBED
     ": GOTO 750
650  IF A(K) = TX + CX(3) AND B(K
     ) = TY + CY(3) THEN ZX$ = "R
     EFLECTED": GOTO 830
660  IF A(K) = TX + CX(4) AND B(K
     ) = TY + CY(4) THEN ZX$ = "R
     EFLECTED": GOTO 880
670  NEXT K
680 X = TX:Y = TY
690  IF X = 0 OR X = 9 OR Y = 0 OR
     Y = 9 THEN 710
700  GOTO 440
710 S = (S + 2) -  INT ((S + 2) /
     4) * 4
720  IF S = 0 THEN S = 4
730  IF ZX$ = "" THEN ZX$ = "PASS
     ED THROUGH"
740  FLASH : GOSUB 980: PRINT P
750  HTAB 15: VTAB 23: INVERSE : PRINT
     ZX$:ZX$ = "": FOR ZZ = 1 TO
     2000: NEXT ZZ: NORMAL
760  HTAB 1: VTAB 19: CALL  - 958
     : PRINT "POINTS = "PT
770  VTAB 22: PRINT "------ -----
     - -------"
775  ONERR  GOTO 780
780  VTAB 21: CALL  - 868: INPUT
     "Q=QUIT F=FIRE G=GUESS (ENTE
     R CHOICE)";PQ$
790  IF PQ$ = "F" THEN  GOSUB 161
     0: GOTO 310
800  IF PQ$ = "Q" THEN  GOSUB 161
     0: GOTO 1200
810  IF PQ$ = "G" THEN 1250
820  GOTO 780
830 TT = 3: GOSUB 1670
840  GOSUB 1090
850  IF  ABS (XD) = 1 THEN P = TY
     : GOTO 680
860 P = TX
870  GOTO 680
880 TT = 1: GOSUB 1670
890  GOSUB 1090
900  IF  ABS (XD) = 1 THEN P = TY
     : GOTO 680
910 P = TX
920  GOTO 680
930 TT = 2: GOSUB 1670
940  GOSUB 1090
950  GOTO 680
960  REM
970  REM    **CALCULATE HOLE POSI
     TION**
980  ON S GOTO 990,1010,1030,1050

990  VTAB 5: HTAB (2 * P + 8)
1000  GOTO 1060
1010  HTAB 26: VTAB (P + 5)
1020  GOTO 1060
1030  VTAB 14: HTAB (2 * P + 8)
1040  GOTO 1060
1050  VTAB (P + 5): HTAB 8
1060  RETURN
1070  REM
1080  REM    **GET MOTION INCREME
     NTS**
1090  ON S GOTO 1100,1120,1140,11
     60
1100 XD = 0:YD = 1
1110  GOTO 1170
1120 XD =  - 1:YD = 0
1130  GOTO 1170
1140 XD = 0:YD =  - 1
1150  GOTO 1170
1160 XD = 1:YD = 0
1170  RETURN
1180  REM
1190  REM    **PRINT ALL ATOMS**
1200  FOR K = 1 TO N
1210  HTAB (2 * A(K) + 8): VTAB (
     B(K) + 5): PRINT "*"
1220  NEXT K
1230  VTAB 20: HTAB 1: CALL  - 95
     8: VTAB 21: PRINT "THIS IS W
     HERE THE ATOMS WERE!": FOR Z
```



```
     Z = 1 TO 2000: NEXT : GOTO 1
     440
1240  REM
1250  REM    **ENTER GUESS AND CH
     ECK**
1255  ONERR  GOTO 1260
1260  VTAB 20: CALL  - 958
1270  NORMAL : VTAB 21
1280  INPUT "ENTER ROW ";R
1290  PRINT
1295  ONERR  GOTO 1300
1300  VTAB 23: INPUT "ENTER COLUM
     N ";C
1310  FOR W9 = 1 TO CO
1320  IF C = C(W9) AND R = D(W9) THEN
     1410
1330  NEXT W9
1340  FOR K = 1 TO N
1350  IF A(K) = C AND B(K) = R THEN
      INVERSE : HTAB 22: VTAB 22:
      PRINT "CORRECT": FOR EE = 1
      TO 3000: NEXT :CO = CO + 1:
     PT = PT +  INT (500 * DI): GOTO
     1370
1360  GOTO 1390
1370  HTAB (2 * A(K) + 8): VTAB (
     B(K) + 5): NORMAL : PRINT "*
     "
1380 C(CO) = C:D(CO) = R: NORMAL
     : GOSUB 1610: HTAB 1: VTAB 2
     0: CALL  - 958: GOTO 1420
1390  NEXT K
1400  INVERSE : HTAB 22: VTAB 22:
      PRINT "NOT CORRECT": FOR ZZ
      = 1 TO 3000: NEXT : NORMAL

1410  HTAB 1: VTAB 20: CALL  - 95
     8: GOSUB 1610:PT = PT - 100 *
     DI: GOTO 760
1420  IF CO <  > N THEN 760
1430  VTAB 21: HTAB 1: CALL  - 95
     8: VTAB 21: PRINT "YOU GUESS
     ED THEM ALL!!"
1440  VTAB 19: PRINT "POINTS = "P
     T: HTAB 1: VTAB 22
1445  ONERR  GOTO 1450
1450  PRINT : INPUT "PLAY ANOTHER
      GAME?  Y OR N ?";A$: IF  LEFT$
     (A$,1) = "Y" THEN  CLEAR : GOTO
     90
1460  FOR ZZ = 1 TO 1000: NEXT : HOME
     : END
1470  GOTO 310
1480  REM
1490  REM   *GET FIRING COORDINAT
     ES**
1500  ON S GOTO 1510,1530,1550,15
     70
1510 X = P:Y = 0
1520  GOTO 1580
1530 X = 9:Y = P
1540  GOTO 1580
1550 X = P:Y = 9
1560  GOTO 1580
1570 X = 0:Y = P
1580  RETURN
1590  REM
1600  REM    **PRINT HOLE NUMBERS
     **
1610  HTAB 10: VTAB 5: FOR X = 1 TO
     8: PRINT X" ";: NEXT X: PRINT
     : VTAB 6
1620  FOR X = 1 TO 8: HTAB 8: PRINT
     X;: HTAB 26: PRINT X: NEXT X
     : VTAB 14: HTAB 10
1630  FOR X = 1 TO 8: PRINT X" ";
     : NEXT X
1640  RETURN
1650  REM
1660  REM    **CHANGE DIRECTION**

1670 S = (S + TT) -  INT ((S + TT
     ) / 4) * 4
1680  IF S = 0 THEN S = 4
1690  RETURN
1700  REM
1710  REM   **DIRECTIONS FOR GAME
     **
1720  HOME
1730  VTAB 10: HTAB 15: PRINT "BL
     ACK HOLE"
1740  VTAB 12: HTAB 19: PRINT "BY
     "
1750  VTAB 14: HTAB 15: PRINT "CO
     MPU HOUSE "
1760  VTAB 24: HTAB 15: INPUT "PU
     SH RETURN TO GO ON..";PP$
1770  HOME : VTAB 12: INPUT "DIRE
     CTIONS Y=YES  N=NO ";PP$
1780  IF  LEFT$ (PP$,1) = "N" THEN
     2440
1790  SPEED= 200: HOME : VTAB 5
1800  PRINT "  THE COMPUTER WILL
     RANDOMLY PLACE": PRINT : HTAB
     10: PRINT "ATOMS IN A BLACK
     HOLE."
1810  PRINT : PRINT "  YOU MUST G
     UESS WHERE THEY ARE!"
1820  PRINT : PRINT "YOU MUST FIR
     E ATOMS INTO THE HOLE AND"
1830  PRINT : HTAB 2
1840  PRINT "THE COMPUTER WILL SH
     OW WHERE THEY"
1850  PRINT : HTAB 16
1860  PRINT "COME OUT."
1870  GOSUB 2460
1880  HOME : VTAB 5: HTAB 5: PRINT
     "WHEN YOU FIRE AN ATOM INTO
```

原子黑洞



```
     THE ": CALL  - 922: PRINT "B
     LACK HOLE, IT WILL BE...."
1890  CALL  - 922: HTAB 15: PRINT
     "ABSORBED": CALL  - 922: HTAB
     15: PRINT "REFLECTED": CALL
      - 922: HTAB 12: PRINT "OR R
     EJECTED."
1900  GOSUB 2460
1910  HOME : PRINT "THE ATOM IS A
     BSORBED IF YOU FIRE": CALL  -
     922: PRINT "DIRECTLY AT AN A
     TOM IN THE HOLE."
1920  VTAB 5: GOSUB 1610: HTAB 14
     : VTAB 9: PRINT "*": INVERSE
     : HTAB 14: VTAB 14: PRINT "3
     "
1930  GOSUB 2470
1940 QQ = 10: GOSUB 2480
1950  FOR K = 1 TO 1000: NEXT
1960  GOSUB 2460
1970  HOME : PRINT "THE ATOM WILL
      PASS STRAIGHT THROUGH": PRINT
     "THE HOLE IF NO ATOM IS IN T
     HE WAY."
1980  VTAB 5: GOSUB 1610: HTAB 18
     : VTAB 9: PRINT "*": INVERSE
     : HTAB 14: VTAB 14: PRINT "
     "
1990  GOSUB 2470
2000 QQ = 6: GOSUB 2480
2010  VTAB 5: HTAB 14: FLASH : PRINT
     "3": NORMAL
2020  FOR K = 1 TO 1000: NEXT
2030  GOSUB 2460
2040  HOME : PRINT "THE ATOM YOU
     FIRE IS REFLECTED": PRINT "T
     O THE LEFT WHEN IT APPROCHES
     ": PRINT "AN ATOM DIAGONALLY
      TO ITS RIGHT."
2050  VTAB 5: GOSUB 1610: HTAB 16
     : VTAB 9: PRINT "*": INVERSE
     : HTAB 14: VTAB 14: PRINT "3
     "
2060  GOSUB 2470
2070 QQ = 10: GOSUB 2480
2080 QQ = 9:RR =  - 1:PP = 13: GOSUB
     2490
2090  FOR K = 1 TO 1000: NEXT : NORMAL
     : GOSUB 2460
2100  HOME : PRINT "THE ATOM YOU
     FIRE IS REFLECTED": PRINT "T
     O THE RIGHT WHEN IT APPROACH
     ES": PRINT "AN ATOM DIAGONAL
     LY TO ITS LEFT."
2110  VTAB 5: GOSUB 1610: HTAB 12
     : VTAB 9: PRINT "*": INVERSE
     : HTAB 14: VTAB 14: PRINT "3
     "
2120  GOSUB 2470
213C QQ = 10: GOSUB 2480
2140 QQ = 25:RR = 1:PP = 15: GOSUB
     2490
2150  FOR K = 1 TO 1000: NEXT : NORMAL
     : GOSUB 2460
2160  HOME : PRINT "THE ATOM YOU
     FIRE IS REJECTED": PRINT "IF
      IT IS AGAINST A WALL AND": PRINT
     "YOU FIRE INTO THE HOLE NEXT
      TO IT."
2170  VTAB 5: GOSUB 1610: HTAB 12
     : VTAB 13: PRINT "*": INVERSE
     : HTAB 14: VTAB 14: PRINT "3
     "
2180  GOSUB 2470
2190  HTAB 14: VTAB 13: PRINT " "
     : FOR J = 1 TO 500: NEXT : NORMAL
     : HTAB 14: VTAB 13: PRINT "
     ": HTAB 14: VTAB 14: FLASH :
     PRINT "3"
2200  FOR K = 1 TO 1000: NEXT : NORMAL
     : GOSUB 2460
2210  HOME : PRINT "WITH MORE THA
     N ONE ATOM IN THE HOLE": PRINT
     "IT GETS INTERESTING."
2220  VTAB 5: GOSUB 1610: HTAB 12
     : VTAB 9: PRINT "*": HTAB 22
     : VTAB 9: PRINT "*": INVERSE
     : HTAB 14: VTAB 14: PRINT "3
     "
2230  GOSUB 2470
2240 QQ = 10: GOSUB 2480
2250 QQ = 20:RR = 1:PP = 15: GOSUB
     2490
2260  FOR K = 11 TO 13: INVERSE :
      HTAB 20: VTAB K: PRINT " ":
      FOR J = 1 TO 500: NEXT : NORMAL
     : HTAB 20: VTAB K: PRINT " "
     : NEXT K: FLASH : HTAB 20: VTAB
     14: PRINT "6"
2270  FOR K = 1 TO 1000: NEXT K: NORMAL
      : GOSUB 2460
2280  HOME : VTAB 7
2290  PRINT "WHEN YOU ARE READY T
     O GUESS WHERE"
2300  PRINT : PRINT "AN ATOM IS H
     IDDEN...."
2310  PRINT : PRINT "YOU WILL BE
     ASKED FOR THE ROW NUMBER."
2320  PRINT : PRINT "(HORIZONTAL)
       ";: SPEED= 100: INVERSE : FOR
     ZZ = 1 TO 20: PRINT " ";: NEXT

2330  GOSUB 2460
2340  HOME : VTAB 7: PRINT "THEN
     YOU WILL BE ASKED FOR..."
2350  PRINT : PRINT "THE COLUMN N
     UMBER FOR THE HIDDEN ATOM."
2360  PRINT : PRINT "(VERTICAL)";
```

原子黑洞

```
     : INVERSE : SPEED= 100: FOR
     ZZ = 11 TO 20: HTAB 13: VTAB
     ZZ: PRINT " ": NEXT : SPEED=
     200: NORMAL
2370  GOSUB 2460
2380  HOME : VTAB 7: PRINT "THE P
     OINT SYSTEM IS..."
2390  VTAB 10: HTAB 15: PRINT "1-
     4 ATOMS"; SPC( 5);"5-8 ATOMS
     "
2400  HTAB 15: PRINT "---------";
      SPC( 5);"---------"
2410  PRINT : PRINT "FIRE INTO HO
     LE";: HTAB 19: PRINT  - 20; SPC(
     9); - 10: PRINT : PRINT "WRO
     NG GUESS";: HTAB 18: PRINT  -
     100; SPC( 8); - 200
2420  PRINT : PRINT "CORRECT GUES
     S";: HTAB 19: PRINT 500; SPC(
     8);1000: GOSUB 2460
2430  HOME : VTAB 10: PRINT "WOUL
     D YOU LIKE TO REPEAT": PRINT
     : PRINT "THE EXPLANATION? Y
     OR N";: INPUT A$: IF  LEFT$
     (A$,1) = "Y" THEN 1790
2440  SPEED= 255: RETURN
2450  REM
2460  VTAB 24: HTAB 15: INPUT "PU
     SH RETURN TO GO ON..";PP$: RETURN

2470  VTAB 20: PRINT "PUSH THE 'F
     ' KEY TO FIRE...";: GET PP$:
      RETURN
2480  FOR K = 13 TO QQ STEP  - 1:
      INVERSE : HTAB 14: VTAB K: PRINT
     " ": FOR J = 1 TO 250: NEXT
     J: HTAB 14: VTAB K: NORMAL :
      PRINT " ": NEXT K: RETURN
2490  FOR K = PP TO QQ STEP RR: HTAB
     K: VTAB 10: INVERSE : PRINT
     " ": FOR J = 1 TO 250: NEXT
     J: NORMAL : HTAB K: VTAB 10:
      PRINT " ": NEXT K
2500  IF QQ = 20 THEN 2540
2510  IF RR = 1 THEN  HTAB 26: GOTO
     2530
2520  HTAB 8
2530  VTAB 10: FLASH : PRINT "5"
2540  RETURN

]
```

改良程式——閃動訊息

讀友雷貫如寄來了一個改良閃動訊息程式（第六期捉虫問題），我們的軟件組主任認爲非常值得爲大家介紹。雖然第六期的捉虫問題經已選出得獎者，但這個改良程式還是有它刊出的價值。

在新程式中，閃動訊息的水平位置可以隨時受到控制（按A、Z鍵），而其游動的速度亦可以用左右箭咀鍵來控制快慢。讀者們可以自己研究程式的妙處，對於日後計編寫程式時，不難沒有好處。□

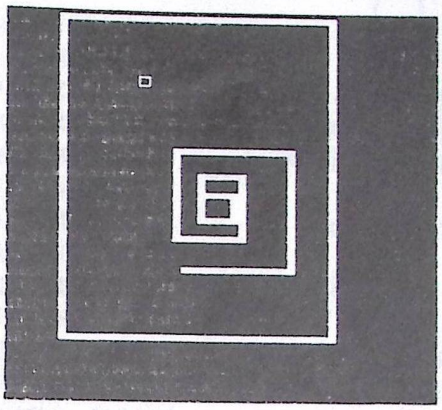
```
LOAD  BUG PROG.LUI CHUN YU

50  HOME : VTAB 10:S = 100:V = 12
     :B$ = "
                         THIS IS A
     TEST. YOU MAY WRITE SOMETHI
     NG FOR YOUR MESSAGE NOT EXCE
     EDING 215 LETTERS AND DON'T
     USE COMMA & COLON. "
100 A$ = B$ + "PLEASE USE THE FOL
     LOWING KEYS : A-UP  Z-DOWN
     <--FAST  -->SLOW  (M)ESSAGE
      (Q)UIT  - C.Y. LUI -": GOTO
     200
150  HOME : PRINT "ENTER YOUR MES
     SAGE : ": PRINT : INPUT "";B
     $:A$ = "
                              " + B$: ON
     B$ = "" GOTO 50
200  HOME : VTAB V
300  HTAB 1: PRINT  LEFT$ (A$,39)
     ;:A$ =  MID$ (A$,2) +  LEFT$
     (A$,1):K =  PEEK ( - 16384):
      IF K < 128 THEN  FOR K = 1 TO
     S: NEXT K:K =  FRE (0): GOTO
     300
330  IF K = 205 THEN  POKE  - 163
     68,0: GOTO 150
340  IF K = 209 THEN  POKE  - 163
     68,0: END
350  IF K = 136 THEN S = S - 30: IF
     S < 30 THEN S = 10
360  IF K = 149 THEN S = S + 30: IF
     S > 200 THEN S = 200
370  IF K = 193 THEN V = V - 1: GOSUB
     400: POKE  - 16368,0: GOTO 2
     00
380  IF K = 218 THEN V = V + 1: GOSUB
     400: POKE  - 16368,0: GOTO 2
     00
390  POKE  - 16368,0: GOTO 300
400  IF V < 1 THEN V = 1
410  IF V > 24 THEN V = 24
420  RETURN
```

蛇之噩夢

這是一個非常抵死好玩的遊戲，莫以爲程式短短，原理簡單而輕視它，玩起上來，比食鬼遊戲還易上癮。

遊戲玩法簡單，閣下只要引領蛇吃去飄忽出現的光點。每吃下一個光點，蛇身會加長一截，而閣下可取得五分。蛇的爬行方向由A、Z←及→（上、下、左、右）鍵控制。如果蛇頭碰著牆壁，撞回自己身體或倒轉而行，閣下便算輸。

光點是會隨時出現及隱沒。看看閣下能吃下多少個光點。



編者按：蛇之噩夢毫無疑問是一個非常有吸引力的遊戲，玩者會愛不惜手的。程式設計相當簡單，並不賣弄任何花巧，可稱得上樸實無華，編者建議，讀者可否嘗試加些花款在這遊戲上，例如使蛇尾偶然擺動。我們歡迎讀者將程式加添花樣，寄來給我們。最佳的改良程式得主，我們將會送給兩期的程式磁碟。有關投稿辦法，可參閱〔投稿需知〕。

本程式在磁碟上的FILENAME是DEVIL SNAKE。□

```
  1 REM **************************
  2 REM *                        *
  3 REM *       DEVIL SNAKE      *
  5 REM *                        *
  6 REM **************************
 10  HOME : HGR
 20  HCOLOR= 3: ROT= 0: SCALE= 1
 30  DIM ST%(800,1)
 40  FOR X = 1 TO 23: READ P: POKE
     767 + X,P: NEXT
 50  DATA  2,0,6,0,13,0,45,53,63,5
     5,45,53,0,45,45,53,54,62,63,
     63,36,36,0
 60  POKE 232,0: POKE 233,3
 70  FOR X = 72 TO 204 STEP 4: DRAW
     1 AT X,2: NEXT
 80  FOR X = 72 TO 204 STEP 4: DRAW
     1 AT X,155: NEXT
 90  FOR Y = 3 TO 154 STEP 3: DRAW
     1 AT 72,Y: NEXT
100  FOR Y = 3 TO 154 STEP 3: DRAW
     1 AT 204,Y: NEXT
110 X = 140:Y = 80:L = 1:N = 1:C =
     1:DI = 1
120 ST%(N,0) = X:ST%(N,1) = Y
130 N = N + 1: IF N = 801 THEN N =
     1
140  XDRAW 1 AT X,Y:CL =  PEEK (2
     34): XDRAW 1 AT X,Y
150  IF CL = 0 THEN 440
160  IF CL <  > 12 THEN  PRINT  CHR$
     (7);:XC = 1:LL = LL + 5:C =
     C + 5: VTAB 23: HTAB 10: PRINT
     "SCORE: ";LL;
170  GOSUB 330
180  DRAW 1 AT X,Y
190  IF C = 0 THEN  XDRAW 1 AT ST
     %(L,0),ST%(L,1):L = L + 1: IF
     L = 801 THEN L = 1
200  IF C > 0 THEN C = C - 1
210 K =  PEEK ( - 16384)
220  IF K < 128 THEN 280
230  POKE  - 16368,0
240  IF K = 193 THEN DI = 1
250  IF K = 149 THEN DI = 2
260  IF K = 218 THEN DI = 3
270  IF K = 136 THEN DI = 4
280  IF DI = 1 THEN Y = Y - 3
290  IF DI = 2 THEN X = X + 4
300  IF DI = 3 THEN Y = Y + 3
310  IF DI = 4 THEN X = X - 4
320  GOTO 120
330  IF XC > 1 THEN XC = XC - 1: GOTO
     430
340  IF XC = 1 THEN  XDRAW 2 AT X
     X,YY:XC = 0
350 RR =  INT ( RND (1) * 10)
360  IF RR <  > 0 THEN 430
370 XX =  INT ( RND (1) * 123) +
     76
380 YY =  INT ( RND (1) * 151) +
     3
390  XDRAW 2 AT XX,YY:CL =  PEEK
     (234): XDRAW 2 AT XX,YY
400  IF CL <  > 18 THEN 370
410  DRAW 2 AT XX,YY
420 XC = 50
430  RETURN
440  VTAB 23: HTAB 1: PRINT "CRAS
     H"; CHR$ (7); CHR$ (7); CHR$
     (7); CHR$ (7): PRINT "PLAY A
     GAIN?";
450  VTAB 24: HTAB 13: GET YN$: IF
     YN$ = "Y" THEN  RUN
460  IF YN$ = "N" THEN  TEXT : HOME
     : END
470  GOTO 450
```

□

自動三角函數曲線繪劃系統

大家都接觸過許多有關三角函數的曲綫繪劃程式，一般都相當簡單。類似本系統所擁有的特色相信會很少遇過。

本系統軟件可任意製定X—Y軸的分格單位數值和比例。而且非常之有彈性，它容許：

（1）X—Y軸的數值可由1E－38至1E＋38 範圍。
（2）任意選擇X，Y軸的濶度和高度。
（3）在螢幕上不同的位置擺放曲綫圖。
（4）在一個畫面上顯示繪劃多個曲綫圖。

系統的特點是可以讓用家自行輸入資料來決定如何繪劃函數曲綫。相信對一般教育性或科學性用途會有很大的幫助。

簡單的說，本系統可分三部份：主索引、輸入資料表，及高解像頁顯示。大家可透過表（1）中所列出的指令鍵來控制整個系統。

按鍵	功能
M	顯示主索引
S	顯示參數變換表
G	顯示曲綫圖
P	在新曲綫圖上繪圖
L	在前一個曲綫圖上繪圖
A	取消或保留X—Y軸
ESC	自動調節X—Y軸比例
SPACE	停止／再繼續繪圖
←，→	上下移動游標

表(1)

當 RUN 本系統時，螢幕會出現一條示範的函數Y＝SIN（X）（見圖1）。大家可以任意更改這條函數的輸入。只要按E（參閱表1的指令鍵）便可作更改。輸入函數是放在行號50的地方，只要重新按入行號50，後隨新函數程式即可。更新函數後，按 RETURN 然後再按 RUN〔RETURN〕即可。

系統可以繪劃一些較複雜的函數，不過假如函數值過大，繪圖速度較慢。

輸入新函數後，便可選擇下列的顯示方式：

（1）在新一頁的畫面上用現有的條件來繪劃新曲綫。
（2）在舊一頁的畫面上繪上新曲綫。
（3）重新再輸入條件參數。

條件參數總共有五項：

(A) 任意決定是否連X—Y軸綫及數目繪出。
(B) 指定繪畫的速度。
(C) 選擇曲綫圖的位置及大小。
(D) 選擇X及Y軸的上下限。
(E) 選擇X—Y軸的單位分間綫及距離。

假如要繪畫的曲綫圖是用來作設計用途，則可不用劃出X—Y軸及標誌單位。辦法簡單，只消按A即可。注意輸入新參數時，數目不得多過9位數字。繪劃的速度有1至5個選擇。〔5〕的速度約爲〔1〕的速度十五倍。如果進行複雜的函數繪劃，建議用慢速。一般來說，使用2或3即可。

曲綫圖的大小取決於像素的濶度及長度。X軸來說由10至255，Y軸則由10至169。整個螢幕的顯示是255×169。如果希望在同一X—Y軸座標上繪劃多個曲綫，建議採用細的曲綫，曲綫越細，X—Y軸所能提供的上下限值便越大。表（2）是各項參數輸入表，大家可以用來參考。

X—Y軸上下限

X軸的上下限可自由輸入。數值必須不能超過9個數位（例如12.345678），上限最大爲0至1E＋38，下限爲－1E＋38至0。

假如輸入的數值絕對值是大過 1E＋8或 1E－8，必須要以上列方式輸入。（留意，數值越大，系統執行速度越慢。Y軸情況亦一樣如是。假如重新輸入分格綫，它必須能

參數	限制
速度	1—5
曲綫圖大小	
濶	10至255
高	10至169
曲綫圖位置	
X	任何變數
Y	任何變數
X—軸上下限	
下限	－1E＋38至0
上限	0至1E＋38
Y—軸上下限	
下限	－1E＋38至0
上限	0至1E＋38
軸分格綫	
X—軸及Y—軸	少於或相等於最大的軸距離 例如X—軸範圍是－10至＋5，X軸的分格綫必須<＝10
數目分格	
X—軸及Y—軸	等於或少於最大的軸範圍。

表(2)

整除X—Y軸的上下限（例如X軸是由0至5，則軸上的分格綫必須是0.25，0.5，1.0，2.5，5。（大過軸限的分格綫是不會被系統接納的。）

繪劃曲綫時，假如曲綫是在幕螢界限範圍，則可看到曲綫繪畫的情形。假如超越幕螢範圍，便不能看到繪畫，最後出來的曲綫圖可能是不完整的。

多次繪劃曲線

大家可以在同一畫面上繪出多個曲綫圖。

要作出多曲綫圖，可依下列方法：

（1）選擇第一個函數，及參數條件，然後繪畫（P）。

（2）假如要改變新函數，可按主索引（M）；輸入新函數，可按（E）。

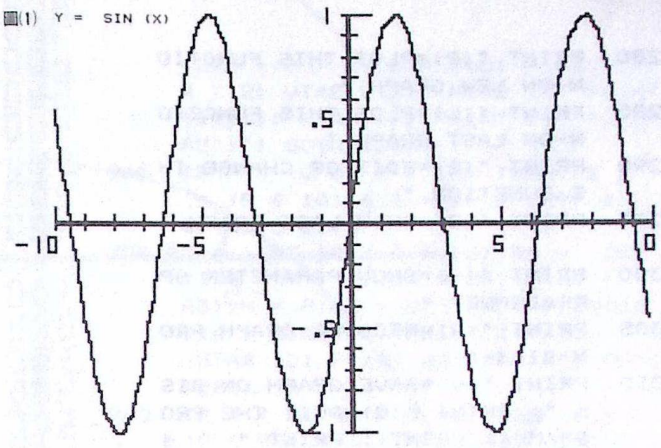
（3）假如新函數曲綫是繪在舊有的X—Y軸上，可按（L）開始。

（4）假如函數是要在不同位置繪劃，可按（S）選擇大小及位置，按（L）開始繪畫。

正如在文章開始提及，本系統可以同時用來作一些曲綫的美術設計，試在沒有X、Y軸情況下，繪出下列函數：

```
1   REM  ************************
2   REM  *                      *
3   REM  * SCIENTIFIC FUNCTION*
4   REM  *       PLOTTER        *
5   REM  *  BY COMPU HOUSE      *
6   REM  *                      *
7   REM  ************************
10  LOMEM: 24576:X = 1.23456789
20  ONERR  GOTO 2830: GOSUB 50
30  GOTO 120
40  REM  **CURRENT FUNCTION **
50 Y =  SIN (X)
60  RETURN
120 JX = 1:JY = 1: GOSUB 50
130 T$(0) = "...":T$(1) = "......
    ......":SP$ = "
                                "
    :E$ = "          ": REM  T$(
    1)=12 PERIODS, SP$=40 SPACES
    , E$=10 SPACES
140  DIM P$(12),P(12),V(12):AX =
    0:AU = 0:CS = 3:CP = 0:IE =
    2
150 AX$(0) = "YES":AX$(1) = " NO"
    :AU$(0) = "OFF":AU$(1) = " N
    O":IE$(0) = " ":IE$(1) = "TO
    O SMALL":IE$(2) = "TOO LARGE
    ":IE$(3) = "TOO LONG"
160  IF  PEEK (31999) <  > 10 THEN
     GOSUB 2780: POKE 31999,10: GOSUB
    1290: GOTO 200
170  FOR A = 0 TO 4:P(A) =  PEEK
    (32000 + A):P$(A) =  STR$ (P
    (A)): NEXT A: FOR A = 5 TO 1
    2:PA = 31990 + 3 * A:P(A) =
    (100 *  PEEK (PA) +  PEEK (P
    A + 1)) * 10 ^ (100 -  PEEK
    (PA + 2)):P(A) =  - 1 ^ (A =
    5 OR A = 7) * P(A):NI = P(A)
    : GOSUB 2370
180  IF CH = 0 THEN P$(A) = NI$:P
    (A) =  VAL (P$(A)): NEXT : GOTO
    200
190 P$(A) =  STR$ (P(A)): NEXT
200  GOSUB 3080
210 V(0) = 6:V(1) = 7:V(2) = 8:V(
    3) = 9:V(4) = 10:V(5) = 12:V
    (6) = 13:V(7) = 15:V(8) = 16
    :V(9) = 17:V(10) = 18:V(11) =
    19:V(12) = 20
220 RS =  INT (10 / (PS ^ 1.7) *
    120)
230 O2 = OY - H / 2:O1 = OX + W /
    2:R1 =  INT (W / 2 + 2):R2 =
     INT (275 - W / 2):R3 =  INT
    (H / 2 + 10):R4 =  INT (178 -
    H / 2):R4 = R4 + (R4 < R3): GOTO
    240
240 N$ = "": INVERSE : TEXT : HOME
    : HTAB 8: PRINT "SCIENTIFIC
    FUNCTION PLOTTER": NORMAL
250  VTAB 3: HTAB 1: PRINT "THIS
    PROGRAM WILL PLOT ANY FUNCTI
    ON      INPUT ON LINE '50'...
    .LINE '50' NOW      HOLDS THE
     FOLLOWING FUNCTION: "
260 DS = 0: VTAB 7: HTAB 1: LIST
    50
270  VTAB 12: HTAB 1: INVERSE : PRINT
    "CHOOSE:": NORMAL
```

圖(1) Y = SIN (X)



Y＝COS （X）

Y＝TAN （X）

Y＝1／COS（X）

本程式佔用了第二頁高解像下的11996 BYTES ，留有 2339 BYTE 作額外附加。

系統可改爲能在同一時間繪劃多個函數曲綫。大家不妨想想如何做法！然後寄來給我們，好讓其他讀者分享。

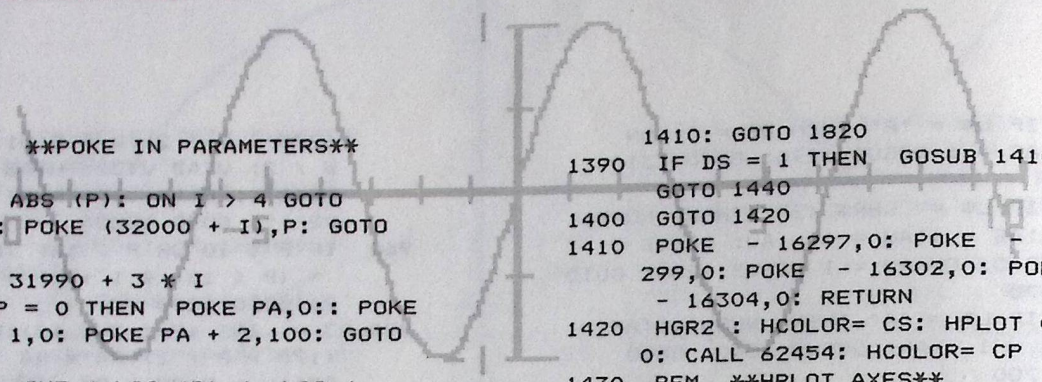
自動三角函數曲線繪劃系統

```
280  PRINT "(P)=PLOT THIS FUNCTIO
     N-ON NEW GRAPH "
285  PRINT "(L)=PLOT THIS FUNCTIO
     N-ON LAST GRAPH "
290  PRINT "(E)=EDIT OR CHANGE TH
     E FUNCTION "
295  PRINT "(G)=SHOW LAST GRAPH "

300  PRINT "(S)=SHOW PARAMETER SP
     READSHEET "
305  PRINT "(R)=RECOVER GRAPH FRO
     M DISK "
310  PRINT "(V)=SAVE GRAPH ON DIS
     K ": PRINT "(Q)=QUIT THE PRO
     GRAM ": PRINT : PRINT "    *
     * COMPU HOUSE ** "
320  IF  PEEK ( - 16384) > 127 THEN
      GET N$: POKE  - 16368,0:T =
     1
330  IF N$ = "Q" THEN  HOME : END

340  IF N$ = "P" THEN  ON AS = 1 GOSUB
     1290: GOTO 1310
350  IF N$ = "L" THEN DS = 1: GOTO
     1310
360  IF N$ = "E" THEN 420
370  IF N$ = "G" THEN  GOSUB 1410
     : GOTO 2060
380  IF N$ = "S" THEN 470
390  IF N$ = "R" THEN 3130
400  IF N$ = "V" THEN 3220
410  GOTO 320
420  HOME : INVERSE : HTAB 8: PRINT
     "EDIT OF CURRENT FUNCTION ":
      NORMAL
430  PRINT : PRINT : PRINT "TO CH
     ANGE CURRENT FUNCTION SHOWN
     BELOW, EARSE AS NEEDED WITH
     <SPACE BAR>, TYPE DIRECTLY O
     VER PARTS TO BE CHANGED, AND
      PASS OVER PARTS NOT TO BE C
     HANGED WITH  '-->' KEY. "
440  VTAB 20: PRINT "REMEMBER TO
     TYPE: ";: FLASH : PRINT "'RU
     N <RETURN>' ": NORMAL
450  POKE 32,2: VTAB 10: LIST 50:
      POKE 32,0: VTAB 10: HTAB 1
460  END
470  REM **SET UP SPREADSHEET **
480  TEXT : HOME : VTAB 1: HTAB 6
     : PRINT "SELECTION OF PLOT C
     ONDITIONS"
490  INVERSE : VTAB 1: HTAB 1: LIST
     50
500 AS = 0
510  VTAB 5: HTAB 1: PRINT "AXIS(
     SWITCH=A)";: NORMAL : PRINT
     AX$(AX);: HTAB 20: INVERSE :
      PRINT "AUTO(SWITCH=<ESC>)";
     : NORMAL : PRINT AU$(AU)
520  VTAB 6: HTAB 1: INVERSE : PRINT
     "SPEED(1-5)";: NORMAL : PRINT
     "..................."
530  VTAB 7: HTAB 1: INVERSE : PRINT
     "GRAPH SIZE";: NORMAL : PRINT
     "*WIDTH(10-255)...."
540  VTAB 8: HTAB 10: PRINT "*HEI
     GHT(10-169)..."
550  VTAB 9: HTAB 1: INVERSE : PRINT
     "GRAPH LOC";: NORMAL : PRINT
     "*X("R1"-"R2").........."
560  VTAB 10: HTAB 10: PRINT "*Y(
     "R3"-"R4")......."
570  INVERSE :: VTAB 11: HTAB 1: PRINT
     "X-AXIS LIMITS:": NORMAL : VTAB
     12: HTAB 3: PRINT "*LOWER...
     ..............."
580  VTAB 13: HTAB 3: PRINT "*UPP
     ER....................."
590  INVERSE : VTAB 14: HTAB 1: PRINT
     "Y-AXIS LIMITS:": NORMAL : VTAB
     15: HTAB 3: PRINT "*LOWER...
     ..............."
600  VTAB 16: HTAB 3: PRINT "*UPP
     ER....................."
610  INVERSE : VTAB 17: HTAB 1: PRINT
     "TIC INTV:";: NORMAL : PRINT
     "*X-AXIS..........."
620  VTAB 18: HTAB 10: PRINT "*Y-
     AXIS..........."
630  INVERSE : VTAB 19: HTAB 1: PRINT
     "NUM INTV:";: NORMAL : PRINT
     "*X-AXIS..........."
640  VTAB 20: HTAB 10: PRINT "*Y-
     AXIS..........."
650  FOR I-I = 0 TO 12: GOSUB 3110
     : NEXT II: NORMAL
660  GOSUB 2960
670  VTAB 22: HTAB 1: PRINT  CHR$
     (91)"P]=PLOT  " CHR$ (91)"<-
     -]=UP " CHR$ (91)"-->]=DOWN
     " CHR$ (91)"M]=MENU"
680  HTAB 1: PRINT  CHR$ (91)"L]=
     PLOT ON LAST GR " CHR$ (91)"
     G]=SHOW LAST GR "
690 I = 5 * (E > 0) * (X <  > X1)
     + 6 * (E > 0) * (X = X1):HZ
     = 30:T = 0
700  VTAB V(I): HTAB HZ: GET L$
710  IF L$ =  CHR$ (8) THEN I = I
     - 1 + 13 * (I = 0): GOTO 70
     0
720  IF L$ =  CHR$ (21) THEN I =
     I + 1 - 13 * (I = 12): GOTO
     700
```

自動三角函數曲線繪劃系統

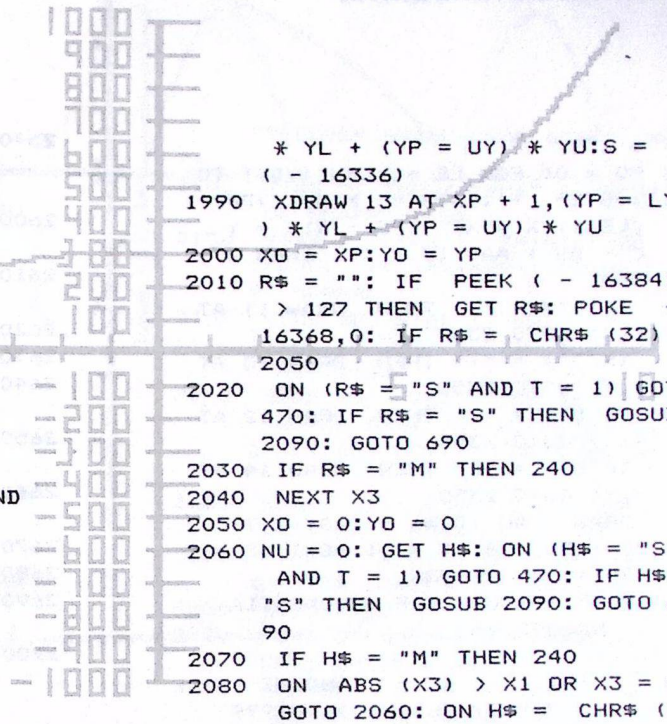
```
730  IF L$ = "P" THEN DS = 0: ON
     AS = 1 GOSUB 1290: GOTO 1310

740  IF L$ =  CHR$ (27) THEN  NORMAL
     :AS = 1:AU = 1 - AU: GOSUB 3
     040: ON AU = 1 GOSUB 1280: GOTO
     70
750  IF L$ = "A" THEN  NORMAL :AX
      = 1 - AX: GOSUB 3040: GOTO
     700
760  IF (L$ =  > "0" AND L$ <  =
     "9") OR L$ = "-" OR L$ = "."
      OR L$ = "+" THEN  GOSUB 810

770  IF L$ = "M" THEN  ON AS = 1 GOSUB
     1290: GOTO 240
780  IF L$ = "G" THEN  GOSUB 1410
     : GOTO 2060
790  IF L$ = "L" THEN DS = 1: ON
     AS = 1 GOSUB 1290: GOTO 1310

800  GOTO 700
810  REM  **INPUT VARIABLES**
820 HL = 31: IF E <  > 0 THEN  VTAB
     23: HTAB 28: PRINT E$
830 V$ = "":N$ = L$
840 V$ = V$ + N$: NORMAL
850  VTAB V(I): HTAB 30: PRINT E$
     ;: HTAB 30: PRINT V$;: HTAB
     HL: GET N$
860  IF N$ =  CHR$ (13) THEN P =
      VAL (V$): ON I + 1 GOTO 920
     ,940,960,980,1000,1020,1040,
     1060,1080,1100,1120,1140,116
     0
870  IF N$ =  CHR$ (8) THEN HL =
     HL - 1 + (HL = 30):V$ =  LEFT$
     (V$, LEN (V$) - 1 + (HL = 30
     )): IF HL = 30 THEN IE = 0: GOTO
     1270
880  IF HL = 37 OR HL = 38 THEN  PRINT
      CHR$ (7)
890  IF (N$ < "0" OR N$ > "9") AND
     N$ <  > "-" AND N$ <  > "E" AND
     N$ <  > "+" AND N$ <  > "." THEN
     850
900 HL = HL + 1: IF HL = 40 THEN
      PRINT  CHR$ (7):IE = 3: GOTO
     1270
910  GOTO 840
920  IF P < 1 OR P > 5 THEN IE =
     (P < 1) * 1 + (P > 5) * 2: GOTO
     1270
930 P =  INT (P):RS =  INT (10 /
     (P ^ 1.7) * 120): GOTO 1260
940  IF P < 10 OR P > 255 THEN IE
      = (P < 10) * 1 + (P > 255) *
     2: GOTO 1270
950 R1 =  INT (P / 2 + 2):R2 =  INT
     (275 - P / 2):W = P:01 = OX +
     W / 2: VTAB V(3): HTAB 13: PRINT
     T$(1);: HTAB 13: PRINT R1"-"
     R2")": GOTO 1260
960  IF P < 10 OR P > 169 THEN IE
      = (P < 10) * 1 + (P > 169) *
     2: GOTO 1270
970 R3 =  INT (P / 2 + 10):R4 =  INT
     (178 - P / 2):R4 = R4 + (R4 <
     R3):H = P:02 = OY - H / 2: VTAB
     V(4): HTAB 13: PRINT T$(1);:
      HTAB 13: PRINT R3"-"R4")": GOTO
     1260
980  IF P < R1 OR P > R2 THEN IE =
     (P < R1) * 1 + (P > R2) * 2:
      GOTO 1270
990 OX = P:01 = OX + W / 2: GOTO
     1260
1000  IF P < R3 OR P > R4 THEN IE
      = (P < R3) * 1 + (P > R4) *
     2: GOTO 1270
1010 OY = P:02 = OY - H / 2: GOTO
     1260
1020  IF P > 0 THEN P =  - P
1030 X2 = P: GOSUB 2960: ON AU GOSUB
     1280:P = X2: GOTO 1260
1040 P =  ABS (P)
1050 X1 = P: GOSUB 2960: ON AU GOSUB
     1280:P = X1: GOTO 1260
1060  IF P > 0 THEN P =  - P
1070 Y2 = P:AU = 0: GOSUB 3040: GOTO
     1260
1080 P =  ABS (P)
1090 Y1 = P:AU = 0: GOSUB 3040: GOTO
     1260
1100 P =  ABS (P):XI = P: IF XI >
     ( ABS (X2) >  =  ABS (X1)) *
      ABS (X2) + ( ABS (X1) >  ABS
     (X2)) *  ABS (X1) THEN 1270
1110 AU = 0: GOSUB 3040: GOTO 126
     0
1120 P =  ABS (P):YI = P: IF YI >
     ( ABS (Y2) >  =  ABS (Y1)) *
      ABS (Y2) + ( ABS (Y1) >  ABS
     (Y2)) *  ABS (Y1) THEN 1270
1130 AU = 0: GOSUB 3040: GOTO 126
     0
1140 P =  ABS (P):NX = P: IF NX >
     ( ABS (X2) >  =  ABS (X1) *
      ABS (X2) + ( ABS (X1) >  ABS
     (X2)) *  ABS (X1) THEN 1270
1150 AU = 0: GOSUB 3040: GOTO 126
     0
1160 P =  ABS (P):NY = P: IF NY >
     ( ABS (Y2) >  =  ABS (Y1)) *
      ABS (Y2) + ( ABS (Y1) >  ABS
     (Y2)) *  ABS (Y1) THEN 1270
1170 AU = 0: GOSUB 3040: GOTO 126
     0
```

自動三角函數曲線繪劃系統



```
1180  REM  **POKE IN PARAMETERS**
1190 P =  ABS (P): ON I > 4 GOTO
     1200: POKE (32000 + I),P: GOTO
     1250
1200 PA = 31990 + 3 * I
1210  IF P = 0 THEN  POKE PA,0:: POKE
     PA + 1,0: POKE PA + 2,100: GOTO
     1240
1220 E1 =  INT ( LOG (P) /  LOG (
     10)) - 3:E2 = E1 + 2:M1 =  INT
     (P / 10 ^ E1 + .5):M2 =  INT
     (M1 / 100):M3 = M1 - M2 * 10
     0
1230  POKE PA,M2: POKE PA + 1,M3:
      POKE PA + 2,100 - E1
1240  IF I = 5 OR I = 7 THEN P =
      - P
1250  RETURN
1260 E = 0:NI = P: GOSUB 2370:P$(
     I) = NI$:P(I) = P: VTAB V(I)
     : HTAB 30: PRINT E$:II = I: GOSUB
     3110: ON AU <  > 1 GOSUB 308
     0:AS = 1: RETURN
1270  PRINT  CHR$ (7): VTAB V(I):
      HTAB 30: INVERSE : PRINT IE
     $(IE);: FOR S = 0 TO 1000: NEXT
     : NORMAL :: HTAB 30: PRINT E
     $:IE = 2: RETURN
1280  GOSUB 2490: FOR II = 7 TO 1
     2: GOSUB 3110: NEXT II: NORMAL
     : RETURN
1290  FOR I = 0 TO 12:P = P(I): GOSUB
     1190:: NEXT I: NORMAL : RETURN
1300  REM  ** CALCULATE LOCATION
     OF GRAPH **
1310 YY = H / (Y1 - Y2):XX = W /
     (X1 - X2)
1320 OX = (OX >  = R1 AND OX <  =
     R2) * OX + (OX < R1) * R1 +
     (OX > R2) * R2:P(3) = OX:P$(
     3) =  STR$ (OX):O1 = OX + W /
     2
1330 OY = (OY >  = R3 AND OY <  =
     R4) * OY + (OY < R3) * R3 +
     (OY > R4) * R4:P(4) = OY:P$(
     4) =  STR$ (OY):O2 = OY - H /
     2
1340 YC = O2 + Y1 * YY:XC = O1 -
     X1 * XX
1350  REM  **CHOOSE PLOTTING MODE
     **
1360  IF DS = 1 AND AX = 1 THEN  GOSUB
     1410: GOTO 1820
1370  IF AX = 1 THEN  HGR2 : HCOLOR=
     CS: HGR2 : HPLOT 0,0: CALL 6
     2454: GOTO 1820
1380  IF DS = 1 AND AS = 0 THEN  GOSUB
      1410: GOTO 1820
1390  IF DS = 1 THEN  GOSUB 1410:
      GOTO 1440
1400  GOTO 1420
1410  POKE  - 16297,0: POKE  - 16
     299,0: POKE  - 16302,0: POKE
      - 16304,0: RETURN
1420  HGR2 : HCOLOR= CS: HPLOT 0,
     0: CALL 62454: HCOLOR= CP
1430  REM  **HPLOT AXES**
1440  FOR L = 1 TO 2
1450  HPLOT O1 - W + JX - 1,YC -
     L + JY TO O1 - 1 + JX,YC - L
      + JY: HPLOT XC - L + JX,O2 +
     JY - 1 TO XC - L + JX,O2 + H
      + JY - 1
1460  NEXT L
1470  REM  **DRAW X-AXIS TICS AND
      NUMBERS**
1480  SCALE= 1: ROT= 0
1490  IF X1 >  ABS (X2) THEN  FOR
     R = X1 TO X2 + X2 / 1000 STEP
      - XI: GOTO 1510
1500  FOR R = X2 TO X1 + X1 / 100
     0 STEP XI
1510 AD = R * XX
1520  IF R = 0 AND Y2 = 0 THEN 15
     50
1530  IF  ABS (R) < X1 / 100 THEN
     1550
1540  DRAW 2 AT XC + AD + JX,YC +
     JY - 2
1550  NEXT R
1560  IF X1 >  ABS (X2) THEN  FOR
     R = X1 TO X2 + X2 / 1000 STEP
      - NX: GOTO 1580
1570  FOR R = X2 TO X1 + X1 / 100
      STEP NX
1580 AD = R * XX: IF  ABS (R) < .
     01 THEN XD = R: GOTO 1600
1590 XD =  INT (10000 * R + .5) /
     10000
1600  IF  ABS (XD) <  = X1 / 1E5 AND
      ABS (Y2) < Y1 / 50 THEN XD =
     0: GOTO 1620
1610  IF  ABS (XD) <  = X1 / 50 THEN
     1630
1620  GOSUB 2100
1630  NEXT R
1640  REM  **DRAW Y-AXIS TICS AND
      NUMBERS**
1650  ROT= 16
1660  IF Y1 >  ABS (Y2) THEN  FOR
     R = Y1 TO Y2 + Y2 / 1000 STEP
      - YI: GOTO 1680
1670  FOR R = Y2 TO Y1 + Y1 / 100
     0 STEP YI
1680 BD = R * YY
1690  IF (R = 0 AND X2 < 0) OR  ABS
```

自動三角函數曲線繪劃系統



```
     (R) < Y1 / 100 THEN 1710
1700  DRAW 2 AT XC - 1 + JX,YC -
     BD + JY
1710  NEXT R
1720  IF Y1 > ABS (Y2) THEN FOR
     R = Y1 TO Y2 + Y2 / 1000 STEP
      - NY: GOTO 1740
1730  FOR R = Y2 TO Y1 + Y1 / 100
      STEP NY
1740 BD = R * YY: IF ABS (R) < .
     01 THEN YD = R: GOTO 1760
1750 YD = INT (10000 * R + .5) /
     10000
1760  IF ABS (YD) < = Y1 / 1E5 AND
      ABS (X2) < X1 / 50 THEN YD =
     0: GOTO 1790
1770  IF ABS (YD) < Y1 / 50 THEN
     1800
1780  IF ABS (YD) < ABS (Y2 / 5
     0) THEN 1800
1790  GOSUB 2230
1800  NEXT R
1810  REM **PLOT FUNCTION **
1820  HCOLOR= CP
1830 ST = ABS (X1 - X2) / RS: ROT=
     0:YU = 02 + H + JY + 11:YL =
     02 + JY - 9:XO = 0:YO = 0
1840 LY = 02:UY = 02 + P(2): REM
      UPPER & LOWER Y-PLOT LIMITS

1850  FOR X3 = X2 TO X1 STEP ST:X
      = X3
1860  GOSUB 50
1870 XP = XC + X3 * XX:YP = YC -
     Y * YY: IF XP < 01 - W OR XP
      > 01 THEN 2040
1880  IF YP > LY AND YP < UY AND
     YO > LY AND YO < UY THEN 197
     0
1890  IF YP < LY AND YO > = UY THEN
     YP = LY: GOTO 1980
1900  IF YP < LY AND YO < = LY THEN
     YP = LY: GOTO 1980
1910  IF YP > UY AND YO > = UY THEN
     YP = UY: GOTO 1980
1920  IF YP > UY AND YO < = LY THEN
     YP = UY: GOTO 1980
1930  IF YP < LY AND YO > LY AND
     YO < UY THEN YP = LY: GOTO 1
     970
1940  IF YP > UY AND YO > LY AND
     YO < UY THEN YP = UY: GOTO 1
     970
1950  IF XO = 0 AND YO = 0 THEN HPLOT
     XP,YP: GOTO 2000
1960  IF YP > UY THEN 2040
1970  HPLOT XO,YO TO XP,YP: GOTO
     2000
1980  DRAW 13 AT XP + 1,(YP = LY)
      * YL + (YP = UY) * YU:S = PEEK
     ( - 16336)
1990  XDRAW 13 AT XP + 1,(YP = LY
     ) * YL + (YP = UY) * YU
2000 XO = XP:YO = YP
2010 R$ = "": IF PEEK ( - 16384)
      > 127 THEN GET R$: POKE -
     16368,0: IF R$ = CHR$ (32) THEN
     2050
2020  ON (R$ = "S" AND T = 1) GOTO
     470: IF R$ = "S" THEN GOSUB
     2090: GOTO 690
2030  IF R$ = "M" THEN 240
2040  NEXT X3
2050 XO = 0:YO = 0
2060 NU = 0: GET H$: ON (H$ = "S"
      AND T = 1) GOTO 470: IF H$ =
     "S" THEN GOSUB 2090: GOTO 6
     90
2070  IF H$ = "M" THEN 240
2080  ON ABS (X3) > X1 OR X3 = 0
      GOTO 2060: ON H$ = CHR$ (3
     2) GOTO 1980: GOTO 2060
2090  POKE - 16303,0: POKE - 16
     300,0: RETURN : REM SWITCH
     TO TEXT
2100  REM ** DRAW X-AXIS NUMBERS
     **
2110 NU = 1
2120  IF XD = 0 THEN 2140
2130  DRAW 2 AT XC + AD + JX,YC +
     JY - 5
2140 NI = XD: GOSUB 2370:XD$ = NI
     $
2150 PO = 0: FOR LE = LEN (XD$) TO
     1 STEP - 1:DI$ = MID$ (XD$
     ,LE,1):X = XC + AD + PO:Y =
     YC + 8 + AA: IF X < 1 THEN 2
     220
2160  IF DI$ = "." THEN DRAW 11 AT
     X,Y: GOTO 2210
2170  IF DI$ = "-" THEN DRAW 13 AT
     X,Y: GOTO 2210
2180  IF DI$ = "+" THEN DRAW 12 AT
     X,Y: GOTO 2210
2190  IF DI$ = "E" THEN DRAW 14 AT
     X,Y: GOTO 2210
2200  DRAW VAL (DI$) + 1 AT X,Y
2210 PO = PO - 6:AA = 0: NEXT LE
2220 DI$ = "": RETURN
2230  REM ** DRAW Y-AXIS NUMBERS
     **
2240  IF X1 < ABS (X2) / 10 THEN
     2270
2250  IF YD = 0 THEN 2270
2260  DRAW 2 AT XC + 5 + JX,YC -
     BD + JY
2270  ROT= 0
2280 NI = YD: GOSUB 2370:YD$ = NI
```

```
$
2290 PO = 0: FOR LE =  LEN (YD$) TO
     1 STEP  - 1:DI$ =  MID$ (YD$
     ,LE,1):X = XC + PO - 8:Y = Y
     C - BD + AA: IF X < 1 THEN 2
     360
2300  IF DI$ = "." THEN  DRAW 11 AT
     X,Y: GOTO 2350
2310  IF DI$ = "-" THEN  DRAW 13 AT
     X,Y: GOTO 2350
2320  IF DI$ = "+" THEN  DRAW 12 AT
     X,Y: GOTO 2350
2330  IF DI$ = "E" THEN  DRAW 14 AT
     X,Y: GOTO 2350
2340  DRAW  VAL (DI$) + 1 AT X,Y
2350 PO = PO - 6:AA = 0: NEXT LE
2360  ROT= 16: RETURN
2370  REM  **ROUND OFF EXPONENTIA
     L  VALUES **
2380 CH = 0
2390  IF (NI >  - 1E + 9 AND NI <
      - .0099999) OR (NI > .00999
     9 AND NI < 1E + 9) OR (NI =
     0) THEN NI$ =  STR$ (NI):CH =
     1: RETURN
2400  IF NI >  = .01 AND NI < 1E +
     9 THEN NI$ =  STR$ (NI): RETURN

2410 XY = NI
2420 N = 0: IF NI < 0 THEN N = 1
2430 AB =  ABS (NI):AB$ =  STR$ (
     AB):LG$ =  RIGHT$ (AB$,3):LG
      =  VAL (LG$)
2440 A1 = AB / 10 ^ LG:A1 =  INT
     (100 / 10 ^ NU * A1 + .5) /
     (100 / 10 ^ NU)
2450 NI$ =  STR$ (A1) + "E" + LG$

2460  IF N = 1 THEN NI$ = "-" + N
     I$
2470 NI =  VAL (NI$)
2480  RETURN
2490  REM  ** CALCULATE AUTO MAX
     & MIN VALUES FOR Y-AXIS **
2500 X = X2: GOSUB 50
2510  IF A = 0 OR A > 0 THEN Y2 =
     0: GOTO 2570
2520  IF A = Y THEN Y2 = A: GOTO
     2540
2530 Y = A: GOSUB 2640:Y2 = YY: IF
      ABS (Y2) > 10 THEN Y2 =  -
      INT ( ABS (Y2))
2540 X = X1: GOSUB 50
2550  IF B = 0 OR B < 0 THEN Y1 =
     0: GOTO 2580
2560  IF B = Y THEN Y1 = B: GOTO
     2580
2570 Y = B: GOSUB 2640:Y1 = YY
2580  GOSUB 2690
2590 P(5) = X2:P(6) = X1:P(7) = Y
     2:P(8) = Y1:P(9) = XI:P(10) =
     YI:P(11) = NX:P(12) = NY
2600  FOR II = 5 TO 12:NI = P(II)

2610  GOSUB 2370:P(II) = NI:P$(II
     ) = NI$
2620  NEXT II: GOSUB 3080
2630  RETURN
2640 YA =  ABS (Y):N =  INT ( LOG
     (YA) /  LOG (10))
2650  IF YA = 10 ^ N THEN YY = YA
     : GOTO 2670
2660 YY = (10 ^ N) *  INT ((YA /
     10 ^ N) + 1)
2670  IF Y < 0 THEN YY =  - YY
2680  RETURN
2690  IF X1 = 0 THEN XI =  ABS (X
     2) / 10: GOTO 2710
2700  IF X1 <  =  ABS (X2) THEN X
     I =  ABS (X2) / 10:X1 = XI *
      INT (X1 / XI + .5)
2710  IF X2 = 0 THEN XI = X1 / 10
     : GOTO 2730
2720  IF X1 >  ABS (X2) THEN XI =
     X1 / 10:X2 = XI *  INT (X2 /
     XI + .5)
2730  IF Y1 = 0 THEN YI =  ABS (Y
     2) / 10: GOTO 2750
2740  IF Y1 <  =  ABS (Y2) THEN Y
     I =  ABS (Y2) / 10:Y1 = YI *
      INT (Y1 / YI + .5)
2750  IF Y2 = 0 THEN YI = Y1 / 10
     : GOTO 2770
2760  IF Y1 >  ABS (Y2) THEN YI =
     Y1 / 10:Y2 = YI *  INT (Y2 /
     YI + .5)
2770 NY = 5 * YI:NX = 5 * XI: RETURN

2780  REM  **LOAD SHAPE TABLE AND
      INITIAL PLOTTING DATA**
2790  FOR A = 768 TO 923: READ D:
      POKE A,D: NEXT : POKE 232,0
     : POKE 233,3
2800  FOR I = 0 TO 12: READ P(I):
     P$(I) =  STR$ (P(I)): NEXT
2810  RETURN
2820  REM  **SYNTAX ERROR IN FUNC
     TION OR DOS ERROR**
2830 ER =  PEEK (222): IF ER > 15
      THEN 2850
2840  TEXT : HOME : VTAB 5: PRINT
     "DISK ACCESS ERROR NO. "ER: VTAB
     7: PRINT "CONSULT THE DOS MA
     NUAL BEFORE PROCEEDING ": GOTO
     2860
2850  TEXT : HOME : VTAB 5: HTAB
     1: PRINT "THE FUNCTION HAS A
      SYNTAX ERROR.": PRINT : PRINT
```

```
      "YOU MUST CORRECT IT BEFORE
      PROCEEDING."
2860   FOR A = 1 TO 3: CALL  - 105
      9: NEXT : VTAB 20: HTAB 8: PRINT
      "PRESS ANY KEY TO CONTINUE":
       GET B$: PRINT : GOTO 240
2870   REM  **ILLEGAL QUANTITY OR
      OVERFLOW ERROR**
2880   POKE 216,0
2890   IF E >  = 2 THEN  RUN
2900   VTAB 21: HTAB 1: PRINT SP$:
       VTAB 22: HTAB 1: PRINT SP$
2910   TEXT : HOME : HTAB 1: VTAB
      21: FLASH : PRINT "ILLEGAL Q
      UANTITY OR  OVERFLOW ERROR":
       INVERSE : HTAB 16:: PRINT "
      TRY AGAIN";: NORMAL
2920   PRINT  CHR$ (7); CHR$ (7); CHR$
      (7)
2930   FOR S = 1 TO 1000: NEXT
2940 E = E + 1
2950   GOTO 670
2960   REM  **CALCULATE Y AXIS LIM
      ITS FOR DISPLAY**
2970   ONERR  GOTO 2870
2980   VTAB 21: HTAB 1: PRINT SP$
2990   FLASH : VTAB 21: HTAB 15: PRINT
      "STAND BY": NORMAL :X = X1: GOSUB
      50:A = Y:B = Y:SP = (X1 - X2
      ) / 25: FOR X = X2 TO X1 STEP
      SP: GOSUB 50: IF B < Y THEN
      B = Y
3000   IF A > Y THEN A = Y
3010   NEXT X
3020   VTAB 21: PRINT SP$: VTAB 21
      : PRINT "CALC Y LIMITS: ";:
       INVERSE : PRINT A" TO "B: NORMAL
3030   RETURN
3040   REM  ** DISPLAY STATUS OF A
      XIS AND AUTO SWITCHES**
3050   VTAB 5: HTAB 15: PRINT AX$(
      AX)
3060   VTAB 5: HTAB 38: PRINT AU$(
      AU)
3070   RETURN
3080 PS = P(0):W = P(1):H = P(2):
      OX = P(3):OY = P(4):X2 = P(5
      ):X1 = P(6):Y2 = P(7):Y1 = P
      (8):XI = P(9):YI = P(10):NX =
      P(11):NY = P(12)
3090   RETURN
3100   REM  **UPDATE SPREADSHEET**

3110   NORMAL :HT = 29 -  LEN (P$(
      II)): VTAB V(II): HTAB 17 +
      8 * (II < 5): PRINT T$(II >
      4);: INVERSE : HTAB HT: PRINT
      P$(II): RETURN
3120   REM  ** BLOAD GRAPH **
3130   HOME : HTAB 9: INVERSE : PRINT
      "RECOVER GRAPH FROM DISK": NORMAL

3140   VTAB 5: INPUT "WHAT IS NAME
       OF GRAPH ?";G$
3150   VTAB 8: HTAB 1: PRINT "YOU
      CHOSE: ";: INVERSE : PRINT G
      $: NORMAL
3160   VTAB 10: HTAB 1: PRINT "OK?
      (Y/N)";: GET Y$: IF Y$ = "N"
       THEN 3160
3170   VTAB 15: HTAB 1: PRINT "PUT
       PROPER DISK IN THE DRIVE AN
      D HIT ANY KEY ";: GET R$: PRINT

3180   FLASH : VTAB 18: HTAB 16: PRINT
      "LOADING": NORMAL
3190   HGR2 : PRINT  CHR$ (4);"BLO
      AD ";G$",A$4000"
3200   GOTO 2060
3220   HOME : INVERSE : HTAB 11: PRINT
      "SAVE GRAPH TO DISK "
3230   NORMAL : VTAB 5: INPUT "WHA
      T IS NAME OF GRAPH?";N$
3240   VTAB 8: HTAB 1: PRINT "YOU
      CHOSE:";: INVERSE : PRINT N$

3250   NORMAL
3260   VTAB 10: HTAB 1: PRINT "OK?
      (Y/N)";: GET Y$: IF Y$ = "N"
       THEN 3220
3270   VTAB 15: HTAB 1: PRINT "INS
      ERT PROPER DISK IN DRIVE AND
       HIT ANY KEY TO SAVE ";: GET
      R$
3280   FLASH : VTAB 18: HTAB 17: PRINT
      "SAVING": NORMAL
3290   PRINT : PRINT  CHR$ (4);"BS
      AVE "N$;",A$4000,L$2000"
3300   GOTO 240
3310   DATA  14,0,30,0,40,0,46,0,5
      6,0,69,0,78,0,90,0,100,0,107
      ,0,119,0,129,0,133,0,141,0,1
      45,0,36,60,63,54,54,54,45,37
      ,36,0,147,34,36,36,36,0,146,
      63,39,36,45
3320   DATA  37,36,63,39,0,59,15,2
      4,8,24,25,53,54,54,62,63,4,0
      ,59,39,36,77,54,54,54,4,0,8,
      24,56,63,54,46,45,54,62,63,4
      ,0,27,45,54,62,63,36,36,36,4
      ,0,146,36
3330   DATA 36,36,63,39,0,32,60,63
      ,54,46,45,54,62,63,36,4,0,14
      6,36,36,36,63,55,54,45,4,0,1
      8,55,37,0,63,63,12,88,49,22,
      38,0,63,63,4,0,59,39,36,45,1
      81,146,58,63,36,4,0
3340   DATA  3,240,169,140,94,-10,
      10,-1,1,1,.1,5,.5
```

收銀機

這是一個模擬收銀機操作的短程式，實用性不算太大，但頗爲有趣。

程式開首會詢問收入是否包括繳付利得稅或有關稅項（在外國顧客購物是要兼付購物稅的），不過在香港則不用而矣。這裡只是方便用家而設。

收入可以分爲現金及信用咭（或支票等），可以計算退還款項。

程式執行後會出現一個MENU

（1）現金收入
（2）信用收入
（3）退還款項
（4）總 收 入

只要按螢幕指示去做便可。

本程式在程式磁碟中的FILENAME是CASHIER。□

```
1 REM ************************
2 REM *                      *
3 REM *        CASHIER       *
5 REM *                      *
6 REM ************************
10  HOME : VTAB (11): HTAB (8):
    INVERSE : PRINT "$$$ THE CASHIER
    $$$": NORMAL
20  FOR I = 0 TO 4000: NEXT
30  DIM X(25): GOSUB 630
40  HOME : PRINT : PRINT  TAB( 14
    );"TYPE OF SALE:": PRINT
50  PRINT "1=CASH 2=CHARGE 3=REFU
    ND 4=TOTALS 5=QUIT";: GET TY
    $:TY =  VAL (TY$)
60  IF TY < 1 OR TY > 5 THEN  PRINT
    : PRINT "SELECT 1-5 ONLY. RE
    ENTER.": FOR I = 0 TO 2000: NEXT
    : GOTO 40
70  ON TY GOTO 80,80,280,420,520
80  REM
90 I = 1:S = 1:TS = 0
100  HOME : PRINT "PRESS RETURN A
    FTER EACH ITEM."
110  PRINT "ENTER 0 FOR LAST ITEM
    .": PRINT
120  PRINT "ITEM #";I;"  ";: INPUT
    X$:X(S) =  VAL (X$)
130  IF X(S) = 0 THEN  GOTO 150
140 I = I + 1:S = S + 1: GOTO 120

150  FOR S = 1 TO I:TS = TS + X(S
    ): NEXT S
160  PRINT : PRINT  TAB(
    8);"TOTAL SALE:   ";TS
170  IF ST$ = "Y" THEN TX =  INT
    (((TS * TP / 100) + .005) *
    100) / 100: PRINT : PRINT  TAB(
    9);"SALES TAX:   ";TX:TS = T
    S + TX: PRINT : PRINT "  TOT
    AL AMOUT DUE:   ";TS
180  IF TY = 2 THEN  GOTO 240
190  PRINT : INPUT "  AMOUNT TEND
    ERED:   ";TD
200  PRINT : PRINT  TAB( 12);"CHA
    NGE: " INT ((TD - TS + .005)
     * 100) / 100
210 IT = IT + I:GT = GT + TS:TT =
    TT + TX
220  PRINT : PRINT "PRESS ANY KEY
     TO CONTINUE": GET Z$
230  GOTO 40
240  REM
250 IT = IT + 1:CT = CT + TS:TT =
    TT + TX
260  PRINT : PRINT "PRESS ANY KEY
     TO CONTINUE": GET Z$
270  GOTO 40
280  REM
290  HOME : PRINT "PRESS RETURN A
    FTER EACH ITEM ."
300  PRINT "ENTER  0 FOR LAST ITE
    M.": PRINT
310 I = 1:S = 1:TS = 0
320  PRINT "ITEM #";I;"  ";: INPUT
    X$:X(S) =  VAL (X$)
330  IF X(S) <  > 0 THEN I = I +
    1:S = S + 1: GOTO 320
340  FOR S = 1 TO 1:TS = TS + X(S
    ): NEXT S
350  PRINT : PRINT "TOTAL OF ITEM
    S: ";TS
360  IF ST$ = "Y" THEN TX =  INT
    (TS * TP + .5) / 100: PRINT
     TAB( 6);"SALES TAX: ";TX: PRINT
    "  TOTAL CREDIT: ";TS + TX
370  PRINT : PRINT "SELECT:1=CASH
     REFUND 2=CREDIT TO ACCOUNT"
    : GET RF$:RF =  VAL (RF$)
380  IF RF = 1 THEN IT = IT - II:
    GT = GT - TS:TT = TT - TX: GOTO
    400
390 IT = IT - 1:CT = CT - TS:TT =
    TT - TX
400  PRINT : PRINT "PRESS ANY KEY
     TO CONTINUE": GET Z$
410  GOTO 40
420  REM
430  HOME : VTAB (3): PRINT  TAB(
    14);: INVERSE : PRINT "SALES
     SUMMARY": NORMAL
440  PRINT : PRINT  TAB( 19);"ITE
    MS: ";IT
450  PRINT :X = GT: GOSUB 530:GT$
     = X$: PRINT  TAB( 14);"CASH
     SALES:"; TAB( TB);GT$
460  PRINT :X = CT: GOSUB 530:CT$
     = X$: PRINT  TAB( 12);"CRED
    IT SALES:"; TAB( TB);CT$
470  PRINT :X = GT + CT: GOSUB 53
    0: PRINT  TAB( 16);"SUBTOTAL
    :"; TAB( TB);X$
480  IF ST$ = "Y" THEN  PRINT :X =
    TT: GOSUB 530:TT$ = X$: PRINT
     TAB( 15);"SALES TAX:"; TAB(
    TB);TT$
490  PRINT :X = GT + CT + TT: GOSUB
    530: PRINT  TAB( 10);"TOTAL
    RECEIPTS:"; TAB( TB);X$
500  PRINT : PRINT : PRINT  TAB(
    10);"PRESS ANY KEY FOR MENU"
    : GET Z$
510  GOTO 40
520  END
530  REM
540 X$ =  STR$ (X)
550  FOR I = 1 TO  LEN (X$)
560  IF  MID$ (X$,I,1) = "." THEN
    J = I
570  NEXT I
580  IF J =  LEN (X$) THEN X$ = X
    $ + ".00"
590  IF J =  LEN (X$) - 1 THEN X$
     = X$ + "0"
600 TB = 33 -  LEN (X$)
610  RETURN
620  END
630  REM
640  HOME : VTAB (11): PRINT "INC
    LUDE SALES TAX?(Y/N) ";: GET
    ST$
650  IF ST$ = "Y" THEN  PRINT : INPUT
    "TAX PERCENTAGE?(0-10):";TP
660  RETURN
```

□

RESET鍵的深入討論

RESET在APPLE 微電腦中是一個特殊的鍵，當你按下 RESET 鍵之後，APPLE 會立即停止任何正在處理的事，而將系統控制權轉移到鍵盤，並按照當時你所使用的APPLE形式回到監督程式系統，整數 BASIC 系統或 APPLESOFT 系統。

有時按 RESET 鍵會產生很多問題，如對正在進行中的 GAME，按下 RESET 鍵之後，有的是彩色畫面即刻消失，並且在螢幕上出現"APPLEII"的字幕，有的是 GAME重新開始而不停止。如果在執行中的會計程式或類似應用程式時，按下 RESET 鍵，可能會把整個事情弄得一團糟，所以對於這個鍵的功能實在有深入瞭解的必要。

APPLE II 的參考手冊告訴我們每次當你按下 RESET 鍵後，APPLE 將執行下列的各項運作。

1、將控制螢幕的軟性開關設定爲文本和低解像度圖形混合模式；
2、將文本顯示窗，設定爲整個螢幕的大小；
3、將輸出顯示點移到最底行；
4、將字元顯示設定爲正常顯示模式；
5、設定 COUT 和 KEYIN 的軟性開關，使鍵盤和螢幕成爲APPLE的標準輸入和輸出元件。
6、將遊戲I／O接頭中的電鈴(ANNUNCIATOR)輸出"0"和1設定爲1，電鈴輸出"2"和3設定爲0；
7、清除鍵盤選通脈衝；
8、關掉所有的擴展I／O元件；
8、最後發出鈴响一"嗶"。

APPLE II 參考手冊附錄C的 AUTOSTART ROM LISTING內的位址 $FFFC 寫着：FFFC：62FA 669 DW RESET。RESET 的動作即由位址$FA62 開始執行，首先我們把其流程以流程圖一簡單的描述出來，然後再做進一步的說明。

1. D≠0清除十進位元數

當觸發 RESET 鍵之後，首先是將狀態暫存器內的十進位數元清除爲0，其作用是要6502的作業情況以二進位方式進行。

2. SETNORM設定正常顯示模式

APPLE II 的文本模式的字元顯示一共有3種，即正常（黑底白字），反白（白底黑字）和正常及反白交換顯示的閃爍形式，那麼在 RESET 之後，如何知道字元是以那一種形式顯示呢？這就要看第零頁位址$32 的值了。若位址$32 的值是$FF ，字元將以正常形式顯示，若值爲$3F 字元，將以反白形式顯示，若值爲$7F，字元將以閃爍形式顯示，在流程圖二可以知道 SETNORN 副程式 RESET之後，將位址$32的值存入$FF，所以字元將以正常形式（黑底白字）顯示。

3. INIT

APPLE II 的螢幕顯示，只有三種形式，即文本模式，低解像度圖形，及高解像度圖形，每種形式顯示都有兩個記憶區作爲顯示資料的來源，也就是說有兩頁。APPLE 記憶體內有八個稱爲軟性開關的記憶位置（$C050～$C057）用來決定螢幕顯示是那種型，那個頁次。

INIT 副程式首先打開 $C056，$C054，$C051三個軟性開關，亦即螢幕顯示設定爲低解像度圖形，首頁區及文本模式，其次決定文字窗的大小。APPLE 第零頁有四個記憶位址$20，$21，$22，$23分別用來設定文字窗最左端的位置、寬度、文字窗頂端和底端。

當在此四個位址存放不同的值，就可以隨心所欲的變化文字窗的位置和大小。位址$20記存此窗最左端位置，範圍爲0至39（$27）；位址$21記存此窗的寬度，範圍爲1至40（$28）；位址$22爲記此窗頂端位置，範圍爲0至22（$16）；位址$23爲記此窗底端位置，範圍爲1至24（$18；INIT 副程式分別將位址$20至$30的值設定爲$00，$28，$00，$18亦即設定文字窗爲全螢幕大小，

INIT 副程式接着執行基綫位址計算（BASCALC）。其作用是將螢幕垂直座標換成對應記憶體位置，而此記憶體位置是由零頁位址$28（BASL）及$29（BASH）所構成。在文本狀態時 BASH 的範圍爲$40～$07；BASL 的範圍爲$00～$D0(參閱參考手冊圖一 MAP OF THE TEXT SCREEN BASCALC 所計算出來的 BASL，BASH 值即爲字元印出的位置，在RESET之後 INIT 副程式設定垂直座標 CV 爲$17，其計算結果 BASL 值爲$DO，BASH值爲07，此表示爲甚麼 RESET之後，游標會出現在螢幕最底行之原因了。改變 CV 的值，游標出現的位置亦會隨之改變（見流程圖三）。

4. SETVID、SETKBD 設定螢幕輸出、鍵盤輸入

此兩副程式的執行在於將第零頁位址$36，$37及$38，$39分別設定爲$F0，$FD及$1B，$FD，位址$36，$37的記憶位址組稱爲字元輸出開關（CHARACTER OUTPUT SWITCH）簡稱 CSW，位址$36稱爲：CSWL，$37 稱爲 CSWH，$37稱爲 CSWH， 這對地址裏面所含的值是目前 APPLE 用來做單一字元輸出的副程式的位址，另一對記憶體址$38，$39稱爲鍵盤輸入開關（KEYBOARD INPUT SWITCH）簡稱KSW，位址$38稱爲 KSWL，$39稱爲 KSWH。在這位址裏面所含的值是目前 APPLE 用來做單一字元輸入的副程式的位址。

在位址$36，$37存入 $FDF0，這是 COUT 副程式

起始位置，位址$38，$39存入$FD1B ，這是KEYIN副程式的起始位置，所以 RESET 之後，鍵盤就是APPLE 的標準輸出及螢幕標準輸入。

5.觸發開關

遊戲I／O接頭中的電鈴輸出的四對軟性開關所對應的記憶體位址爲：

```
$C058 CLEAR AN#0 OUTPUT
$C059 SET   AN#0 OUTPUT
$C05A CLEAR AN#0 OUTPUT
$C05B SET   AN#1 OUTPUT
$C05C CLEAR AN#2 OUTPUT
$C05D SET   AN#2 OUTPUT
$C05E CLEAR AN#3 OUTPUT
$C05F SET   AN#3 OUTPUT
```

若使用到某一對的第一個位址，那麼就是將此對的電鈴輸出關掉（OFF），若使用到這一對的第二個位址，那麼就是將這個電鈴輸出打開（ON），RESET 之後是將電鈴輸出 AN#0，AN#1，AN#2，AN#3 打開。

記憶位址 $CFFF 是個非常特殊的位置，每個週邊咭都認識它，如果使用到這個位置的指令，將使週邊咭上的擴展 ROM不能運作。另記憶位址$C010是鍵盤選通脈衝開關，RESET之後，所有的擴展I／O元件會被關掉，鍵盤選通脈衝會被清除。

6. BELL

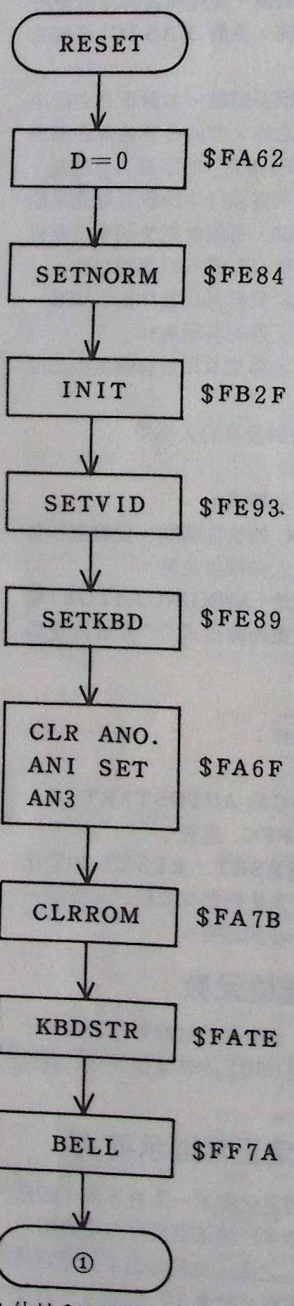
APPLE主機內有個8Ω的揚聲器，接到 APPLE 的內部綫路，能產生各種不同的聲响，它是由一軟性開關所控制，相對的記憶位址是$C030，RESET之後，發出一聲"嗶"頻率 1 KHZ ，時間0.1秒。

RESET的執行

在 APPLE 執行了以上動作後，接著是判斷系統是熱開機（WARM START)還是冷開機（COLD START），熱開機與冷開機最大差別在熱開機的過程中，APPLE內原有的程式和變數均得予保留，而冷開機則在 RESET 之後，所有的程式和變數都消失了，螢幕畫面上重新出現"APPLE Ⅱ"的字樣。

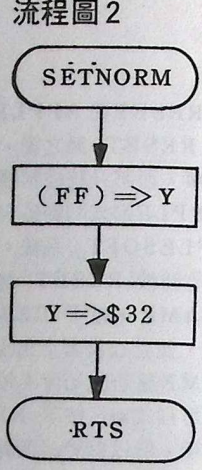
此項判斷的關鍵在檢查第三頁位址$3F2，$3F3，$3F4的值，$3F2 和 $3F3 兩位址是軟體程式進入點的位置向量，當按 RESET 後，程式即指向 $3F2，$3F3 所指的位址去執行。判斷時是檢查開機準備（POWER—ON INITIALIZATION）的動作是否已完成了，亦即位址 $3F3 的值與常數A5做 EOR（EXCLUSIVE OR）的運算所得的結果與位址 $3F4 的值比較，若兩值相等，則表示開機準備完成系統爲熱開機，否則爲冷開機，請看流程圖四。

流程圖1



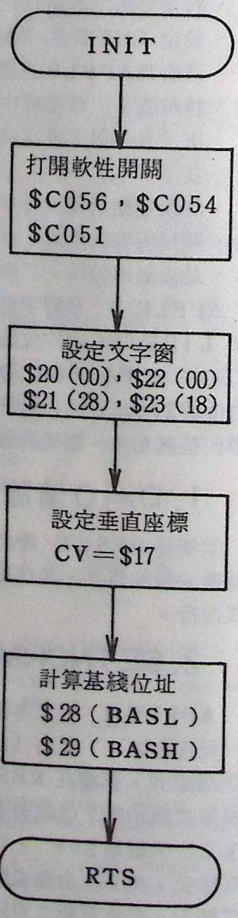
註：位址$FA62，$FE84……等表示該副程式的起始點

流程圖2



註：Y：Y指示暫存器

流程圖3



RESET 鍵的深入討論

若系統判斷爲 COLD START 時，馬上執行開機準備的工作——清除螢幕設定位址$040F～$0416之值爲"APPLE Ⅱ"，並顯示在螢幕及設定位址$3F2，$3F3，$3F4 的值爲 $00，$E0，$45。位址$040F～$0416是在文字顯示幕的第一列16～23的位置，即是開機時"APPLE Ⅱ"字元顯示的位置，流程圖五指出它的設定情形。

由流程圖五可看出它將位址$FB10～$FB09的值存入位址$0416～$040F內，$FB10～$FB09是ROM的記憶體位址，它們的值DB，DD，A0，C5，CC，D0，D0，C1 是燒在 ROM內的 $3F2～$3F4 的值，亦模仿上面方式，由 $FB01～$FAFF的值得來的。

然後檢查 I／O SLOT，從 SLOT 7 到 SLOT 1，看看有沒有磁碟機控制咭，若有便進入 DOS，若無則設定$3F2 的值爲 $03，然後跳到位址$E000，即進入 APPLESOFT 去執行。

若系統判斷爲 WARM START，則檢查$3F2，$3F3的值是否存入$00，$E0，若是則與上述一樣，設定$3F2 的值爲$03，然後跳到$E000去執行，若$3F2，$3F3 的值不是存入 $00，$E0，則系統將轉到 $3F2，$3F3 所指的位址去執行。

前面曾提到過 $3F2，$3F3 的位址是指出RESET後，APPLE 所欲跳往的位址，如果是 $03 和 $E0則進入 BASIC。若是 $69和 $FF，則進入監督程式，所以在你指定好 $3F3，$3F2 的值後利用 POWER—UP 副程式（起始位址 $FB6F）或CALL—1169 來計算$3F4的值，以後只要你按 RESET，系統就會進入你所指定的位址了。

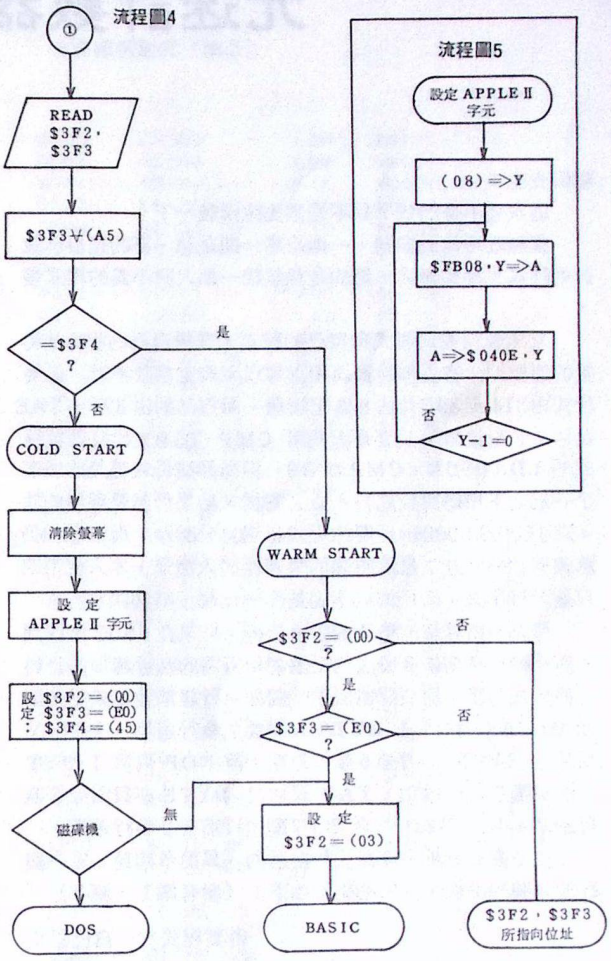
瞭解了 RESET 這個特殊鍵的功能後，相信這對你的程式設計技巧將有所幫助，例如我們可利用 ON RESET GO TO……來保護程式，使程式在 RESET 後，不被取消而能繼續執行，最後，我們舉兩個簡單的例子作爲本文的結論。

例1：按RESET鍵之後，要進入監督程式系統可執行以下程式。

```
10  POKE 1010,105
20  POKE 1011,255
30  POKE 1012,90
40  CALL  - 151: REM  ENTER MONITOR
50  END
```

例2：按RESET鍵之後，要跳到某一指定位址時可執行以下的程式：

```
10  REM  AD IS ADDRESS OF
11  REM  ROUTINE TO RECIVE
12  REM  CONTROL AFTER RESET
20  POKE 1010,AD: REM  SET LOW BYTE
30  POKE 1011,AD: REM  SET HI BYTE
40  CALL  - 1169: REM  SET 1012
```



這裡所附的另一個例子（程式A）是當 RESET 鍵被按下時，程式仍然繼續會執行，但閣下可將1000行輕易地改爲 NEW，LIST或PR#6 以適應閣下的需求。

有很多情形下，市面上的軟件在執行時是不可以按 RESET 鍵的，若不愼按下時會令系統重新啓動，它們是採用以下的方法來達成：

FOR X=1011 TO 1015：POKE X，0：NEXT 或在監督程式內：

*3F3：00 00 00 00 00 00

```
20  ONERR  GOTO 1000
25  POKE 40286,35: POKE 40287,216

30 X = X + 1
40  PRINT X;" ";: IF X < 400 THEN
    30
50  POKE 40286,60: POKE 40287,212
    : END
1000  PRINT : PRINT "START COUNTI
    NG...";:X = 0: GOTO 30
```

光速計數器

編輯先生：

這次並不是鬥快，只不過想先生指教一下。

從報販中取到貴刊，一則是喜一則是怒，喜的是自己設計的程式也算是優異，怒的是竟然較一個大同小異的程式慢。

當比較得獎的程式和自己的程式，發覺自己的程式比得獎的還要短。省去清除畫面和置零以至設定常數不計，得獎程式由314至39E共佔8A記憶體，而自己的由32F至3AE共佔7F記憶體，主要是因爲用 CMP 05BX代替得獎程式的 LDA 05BX，CMP # B9。但爲甚麼仍較得獎的慢呢？一怒之下把得獎程式打入機去測試，結果仍然是得獎的快。以100,000,000計得獎的程式超過11分30秒，而自己的仍然超過12分（註：量度時間的準確性因人而異，本人使用的只是計秒手表，而目的只不過是作一比較，請勿誤會）。

百思不得其解惟指逐個指令分析，結果在6502冊找到一些有關每一個指令所需的記憶體和所需的執行週期的資料。終於找到了，原來記憶體和時間即執行週期並不成反比例如 BREAK 只佔 1 BYTE 却需要7執行週期，RTS 又只佔 1 BYTE 却需要6執行週期，但 NOP 只佔 1 BYTE 只需2執行週期，TAY 只佔 1 BYTE 亦只需要2執行週期，至於 BEQ 佔 2 BYTE 也只需要2執行週期。

於是細心分析得獎程式和自己的，爲簡單起見，省去與計數無關的部份，得出的結果如下：（參考圖1，圖2）

		得獎程式	自己程式
個位由0－9共需	（05C0）	56	58
拾位由0－9共需	（05BF）	138	145
佰位由0－9共需	（05BE）	165	145
仟位由0－9共需	（05BD）	165	145
萬位由0－9共需	（05BC）	165	145
拾萬位由0－9共需	（05BB）	165	145
佰萬位由0－9共需	（05BA）	165	145
仟萬位由0－9共需	（05B9）	165	145

〔單位：執行週期〕

因此由0數至100,000,000共需：

（1）得獎程式：716,333,315執行週期

（2）自己程式：741,111,095執行週期

根據計算的結果亦證明得獎的程式較快，兩者先前省去的部份不足1萬週期，因此影响甚微。

把計算的結果如量度的時間作一比較，發覺兩者極爲吻合（710秒／716,333,315週期＝9.91×10^{-7}；733秒／741,111,095週期＝9.89×10^{-7}）；因此證明了執行週期才是決定程式運行時間的多少，而和記憶體是沒有絕對的關係的。

爲了證明自己的理論，可把以上的程式改良如附圖，其執行週期如下（參考圖3，圖4）

個位由0－9共需	（05C0）	54
拾位由0－9共需	（05BF）	118
佰位由0－9共需	（05BE）	136
仟位由0－9共需	（05BD）	136
萬位由0－9共需	（05BC）	136
拾萬位由0－9共需	（05BB）	136
佰萬位由0－9共需	（05BA）	136
仟萬位由0－9共需	（05B9）	136

〔單位：執行週期〕

由0至100,000,000共需 673,111,096 執行週期，而時間是11分9秒，雖然兩個程式的開始部份不同（清除畫面，置0等），但所需時間是一樣的（指分秒計）。

不知道你能否看明我所寫的東西，但無論如何這兩組新改良的程式是較快的。

失敗者

黃贊輝上

編者註：

我們非常高興接到黃贊輝君的來信，他那股努力不懈的研習精神是應該受到贊賞。我們舉辦比賽的目的並不是要大家努力爭取第一，而是希望能透過比賽，激發起大家潛在著的興趣，及彼此交流意見。

黃君今次雖然未能得獎，但他不旦未感到氣餒，相反更去分析研究爲何程式速度會較他人慢的原因，從而加以改進，這種精神是值得我們學習的。事實上，黃君並不算是失敗者，我們相信，讀者在讀完黃君的來信後，必定有所得益！

我們已由第七期開始，開闢一個園地，專供讀者們發表自己的意見，無論是對本刊的任何意見，或自己本人的一些經驗心得，我們都歡迎大家來信。□

光速計數器

圖（1） 朱永昌 得獎程式

```
0300-  20 58 FC   JSR  $FC58
0303-  A9 B0      LDA  #$B0
0305-  A2 09      LDX  #$09
0307-  9D B7 05   STA  $05B7,X
030A-  CA         DEX
030B-  D0 FA      BNE  $0307
030D-  A0 B1      LDY  #$B1
030F-  A2 B0      LDX  #$B0
0311-  4C 17 03   JMP  $0317
0314-  EE BF 05   INC  $05BF  6
0317-  8C C0 05   STY  $05C0  4
031A-  EE C0 05   INC  $05C0  6
031D-  EE C0 05   INC  $05C0  6
0320-  EE C0 05   INC  $05C0  6
0323-  EE C0 05   INC  $05C0  6
0326-  EE C0 05   INC  $05C0  6
0329-  EE C0 05   INC  $05C0  6
032C-  EE C0 05   INC  $05C0  6
032F-  EE C0 05   INC  $05C0  6
0332-  8E C0 05   STX  $05C0  4
0335-  AD BF 05   LDA  $05BF  4
0338-  C9 B9      CMP  #$B9   2
033A-  D0 D8      BNE  $0314  2
033C-  8E BF 05   STX  $05BF  4
033F-  AD BE 05   LDA  $05BE  4
0342-  C9 B9      CMP  #$B9   2
0344-  F0 06      BEQ  $034C  2
0346-  EE BE 05   INC  $05BE  6
0349-  4C 17 03   JMP  $0317  3
034C-  8E BE 05   STX  $05BE  4
034F-  AD BD 05   LDA  $05BD
0352-  C9 B9      CMP  #$B9
0354-  F0 06      BEQ  $035C
0356-  EE BD 05   INC  $05BD
0359-  4C 17 03   JMP  $0317
035C-  8E BD 05   STX  $05BD
035F-  AD BC 05   LDA  $05BC
0362-  C9 B9      CMP  #$B9
0364-  F0 06      BEQ  $036C
0366-  EE BC 05   INC  $05BC
0369-  4C 17 03   JMP  $0317
036C-  8E BC 05   STX  $05BC
036F-  AD BB 05   LDA  $05BB
0372-  C9 B9      CMP  #$B9
0374-  F0 06      BEQ  $037C
0376-  EE BB 05   INC  $05BB
0379-  4C 17 03   JMP  $0317
037C-  8E BB 05   STX  $05BB
037F-  AD BA 05   LDA  $05BA
0382-  C9 B9      CMP  #$B9
0384-  F0 06      BEQ  $038C
0386-  EE BA 05   INC  $05BA
0389-  4C 17 03   JMP  $0317
038C-  8E BA 05   STX  $05BA
038F-  AD B9 05   LDA  $05B9
0392-  C9 B9      CMP  #$B9
0394-  F0 06      BEQ  $039C
0396-  EE B9 05   INC  $05B9
0399-  4C 17 03   JMP  $0317
039C-  8E B9 05   STX  $05B9
039F-  8C B8 05   STY  $05B8
03A2-  60         RTS
```

56

14× 9+12
=138

17×9+12
=165

黃寶煇舊程式（圖2）

```
0300-  A9 A0      LDA  #$A0
0302-  A2 00      LDX  #$00
0304-  9D 00 04   STA  $0400,X
0307-  9D 00 05   STA  $0500,X
030A-  9D 00 06   STA  $0600,X
030D-  9D 00 07   STA  $0700,X
0310-  CA         DEX
0311-  D0 F1      BNE  $0304
0313-  A2 B0      LDX  #$B0
0315-  8E C0 05   STX  $05C0
0318-  8E BF 05   STX  $05BF
031B-  8E BE 05   STX  $05BE
031E-  8E BD 05   STX  $05BD
0321-  8E BC 05   STX  $05BC
0324-  8E BB 05   STX  $05BB
0327-  8E BA 05   STX  $05BA
032A-  8E B9 05   STX  $05B9
032D-  A9 B9      LDA  #$B9
032F-  EE C0 05   INC  $05C0  6
0332-  EE C0 05   INC  $05C0  6
0335-  EE C0 05   INC  $05C0  6
0338-  EE C0 05   INC  $05C0  6
033B-  EE C0 05   INC  $05C0  6
033E-  EE C0 05   INC  $05C0  6
0341-  EE C0 05   INC  $05C0  6
0344-  EE C0 05   INC  $05C0  6
0347-  EE C0 05   INC  $05C0  6
034A-  8E C0 05   STX  $05C0  4
034D-  CD BF 05   CMP  $05BF  4
0350-  F0 06      BEQ  $0358  2
0352-  EE BF 05   INC  $05BF  6
0355-  4C 2F 03   JMP  $032F  3
0358-  8E BF 05   STX  $05BF  4
035B-  CD BE 05   CMP  $05BE
035E-  F0 06      BEQ  $0366
0360-  EE BE 05   INC  $05BE
0363-  4C 2F 03   JMP  $032F
0366-  8E BE 05   STX  $05BE
0369-  CD BD 05   CMP  $05BD
036C-  F0 06      BEQ  $0374
036E-  EE BD 05   INC  $05BD
0371-  4C 2F 03   JMP  $032F
0374-  8E BD 05   STX  $05BD
0377-  CD BC 05   CMP  $05BC
037A-  F0 06      BEQ  $0382
037C-  EE BC 05   INC  $05BC
037F-  4C 2F 03   JMP  $032F
0382-  8E BC 05   STX  $05BC
0385-  CD BB 05   CMP  $05BB
0388-  F0 06      BEQ  $0390
038A-  EE BB 05   INC  $05BB
038D-  4C 2F 03   JMP  $032F
0390-  8E BB 05   STX  $05BB
0393-  CD BA 05   CMP  $05BA
0396-  F0 06      BEQ  $039E
0398-  EE BA 05   INC  $05BA
039B-  4C 2F 03   JMP  $032F
039E-  8E BA 05   STX  $05BA
03A1-  CD B9 05   CMP  $05B9
03A4-  F0 06      BEQ  $03AC
03A6-  EE B9 05   INC  $05B9
03A9-  4C 2F 03   JMP  $032F
03AC-  8E B9 05   STX  $05B9
03AF-  4C 2F 03   JMP  $032F
03B2-  00         BRK
```

58

15＊9＋10
＝145

光速計數器

新改良程式（圖3）黃贊輝

```
0300-  20 58 FC   JSR   $FC58
0303-  A9 B0      LDA   #$B0
0305-  A2 09      LDX   #$09
0307-  9D B7 05   STA   $05B7,X
030A-  CA         DEX
030B-  D0 FA      BNE   $0307
030D-  A2 B0      LDX   #$B0
030F-  A9 B9      LDA   #$B9
0311-  A0 B1      LDY   #$B1
0313-  D0 03      BNE   $0318
0315-  EE BF 05   INC   $05BF  6
0318-  8C C0 05   STY   $05C0  4
031B-  EE C0 05   INC   $05C0  6
031E-  EE C0 05   INC   $05C0  6
0321-  EE C0 05   INC   $05C0  6
0324-  EE C0 05   INC   $05C0  6
0327-  EE C0 05   INC   $05C0  6   54
032A-  EE C0 05   INC   $05C0  6
032D-  EE C0 05   INC   $05C0  6
0330-  8D C0 05   STA   $05C0  4
0333-  8E C0 05   STX   $05C0  4
0336-  CD BF 05   CMP   $05BF  4
0339-  D0 DA      BNE   $0315  2
033B-  8E BF 05   STX   $05BF  4
033E-  CD BE 05   CMP   $05BE  4
0341-  F0 05      BEQ   $0348  2
0343-  EE BE 05   INC   $05BE  6
0346-  D0 D0      BNE   $0318  2
0348-  8E BE 05   STX   $05BE  4
034B-  CD BD 05   CMP   $05BD
034E-  F0 05      BEQ   $0355
0350-  EE BD 05   INC   $05BD
0353-  D0 C3      BNE   $0318
0355-  8E BD 05   STX   $05BD
0358-  CD BC 05   CMP   $05BC
035B-  F0 05      BEQ   $0362
035D-  EE BC 05   INC   $05BC
0360-  D0 B6      BNE   $0318
0362-  8E BC 05   STX   $05BC
0365-  CD BB 05   CMP   $05BB
0368-  F0 05      BEQ   $036F
036A-  EE BB 05   INC   $05BB
036D-  D0 A9      BNE   $0318
036F-  8E BB 05   STX   $05BB
0372-  CD BA 05   CMP   $05BA
0375-  F0 05      BEQ   $037C
0377-  EE BA 05   INC   $05BA
037A-  D0 9C      BNE   $0318
037C-  8E BA 05   STX   $05BA
037F-  CD B9 05   CMP   $05B9
0382-  F0 05      BEQ   $0389
0384-  EE B9 05   INC   $05B9
0387-  D0 8F      BNE   $0318
0389-  8E B9 05   STX   $05B9
038C-  F0 8A      BEQ   $0318
038E-  60         RTS
038F-  00         BRK
0390-  00         BRK
```

12 * 9 + 10 = 118

14. * 9 + 10 = 136

新改良的程式（圖4）黃贊輝

```
0300-  A9 A0      LDA   #$A0
0302-  A2 00      LDX   #$00
0304-  9D 00 04   STA   $0400,X
0307-  9D 00 05   STA   $0500,X
030A-  9D 00 06   STA   $0600,X
030D-  9D 00 07   STA   $0700,X
0310-  CA         DEX
0311-  D0 F1      BNE   $0304
0313-  A2 B0      LDX   #$B0
0315-  8E C0 05   STX   $05C0
0318-  8E BF 05   STX   $05BF
031B-  8E BE 05   STX   $05BE
031E-  8E BD 05   STX   $05BD
0321-  8E BC 05   STX   $05BC
0324-  8E BB 05   STX   $05BB
0327-  8E BA 05   STX   $05BA
032A-  8E B9 05   STX   $05B9
032D-  A9 B9      LDA   #$B9
032F-  A0 B1      LDY   #$B1
0331-  D0 03      BNE   $0336
0333-  EE BF 05   INC   $05BF
0336-  8C C0 05   STY   $05C0
0339-  EE C0 05   INC   $05C0
033C-  EE C0 05   INC   $05C0
033F-  EE C0 05   INC   $05C0
0342-  EE C0 05   INC   $05C0
0345-  EE C0 05   INC   $05C0
0348-  EE C0 05   INC   $05C0
034B-  EE C0 05   INC   $05C0
034E-  8D C0 05   STA   $05C0
0351-  8E C0 05   STX   $05C0
0354-  CD BF 05   CMP   $05BF
0357-  D0 DA      BNE   $0333
0359-  8E BF 05   STX   $05BF
035C-  CD BE 05   CMP   $05BE
035F-  F0 05      BEQ   $0366
0361-  EE BE 05   INC   $05BE
0364-  D0 D0      BNE   $0336
0366-  8E BE 05   STX   $05BE
0369-  CD BD 05   CMP   $05BD
036C-  F0 05      BEQ   $0373
036E-  EE BD 05   INC   $05BD
0371-  D0 C3      BNE   $0336
0373-  8E BD 05   STX   $05BD
0376-  CD BC 05   CMP   $05BC
0379-  F0 05      BEQ   $0380
037B-  EE BC 05   INC   $05BC
037E-  D0 B6      BNE   $0336
0380-  8E BC 05   STX   $05BC
0383-  CD BB 05   CMP   $05BB
0386-  F0 05      BEQ   $038D
0388-  EE BB 05   INC   $05BB
038B-  D0 A9      BNE   $0336
038D-  8E BB 05   STX   $05BB
0390-  CD BA 05   CMP   $05BA
0393-  F0 05      BEQ   $039A
0395-  EE BA 05   INC   $05BA
0398-  D0 9C      BNE   $0336
039A-  8E BA 05   STX   $05BA
039D-  CD B9 05   CMP   $05B9
03A0-  F0 05      BEQ   $03A7
03A2-  EE B9 05   INC   $05B9
03A5-  D0 8F      BNE   $0336
03A7-  8E B9 05   STX   $05B9
03AA-  F0 8A      BEQ   $0336
03AC-  60         RTS
```

〔任意定義鍵盤〕程式公開發表

我們在第六期擬出一個題目「任意鍵盤」，收至多個自編程式。我們在其中揀取出三個較爲特出的作品公開發表給大家參考。其中鄭偉權讀友在程式文字說明方面最具水準，解釋得相當清晰，而董偉業讀友在程式設計方面相當有創作性。各位讀者可參詳比較三個程式的設計。

TUNG WAI YIP

此程式在執行之後，會顯示兩個鍵盤，上面一個是普通鍵，下面一個則是控制字元（以反白字顯示）。

如果被改和代替的鍵都是英文字母，則控制字元和小楷亦會自動改變：例如將A改成B，則按CTRL－ A和a亦會變成CTRL－B和b。但是H、M、U三個鍵的控制字元，則不會因此而改變，因爲它們是分別代表"←"，"RETURN"和"→"鍵。

改變完畢後，按F "ESC"和"Q"鍵，便可「SAVE」入磁碟，以後用時只要「BRUN FILENAME」即可。

＊若想改變"ESC "鍵，則需要連按兩次即可。□

Lum Kin

這程式用法如下：

鍵入RUN，程式便問要更改何字，按入該字，程式便問該字修改爲那個定義鍵，按入後，可按空位格一次繼續修改，否則按N結束。結束後，鍵盤的字已被更改，若要取消定義鍵可按RESET，若要重置定義鍵，鍵入＆便可。

設計原理：

本人將各鍵之ASCII儲入$320至$37F，一段小小的機械語言儲入$300至$31D ，然後將使用者新定義鍵的ASCII儲入$320至$37F，令字鍵更改。

鑑於APPLESOFT "GET" 字串指令中，若鍵入了CTRL－SHIFT－P ，字串的長度便變爲零，而ASC字串便因而導致ILLEGAL QUANTITY ERROR，CTRL－SHIFT－P 因此不能更改，這可算是一條BUG，故程式中100及110行末段加入了IF STRING=" " THEN STRING=CHR$(0)，就是爲了所有鍵盤字都能更改。

接下來的就是改變KSWL（$38，$39）的向量和將＆指令定義。程式就此完結，以後按鍵時，MONITOR或D•O•S•便會跳至$ 300 執行改鍵的工作，其技巧如同貫刊第5期16進制鍵盤，用了查表的方法。

附帶一提的，本程式可適合磁碟及磁帶用家，程式會自動找尋是否有D•O•S•存在，有則改變D•O•S• 的向量，而貫刊第5期的16進制鍵盤不但不適合磁帶用家，而且程式更有問題存在。□

```
1   HOME : FOR S = 0 TO 95: POKE 8
      00 + S,S + 128: NEXT S
2   FOR S = 768 TO 797: READ N: POKE
      S,N: NEXT S
10   PRINT "CHANGING KEY ";: GET A
      $: PRINT A$: IF A$ = "" THEN
      A$ =  CHR$ (0)
11   PRINT "CHANGE TO KEY ";: GET
      B$: PRINT B$: IF B$ = "" THEN
      B$ =  CHR$ (0)
12   POKE  ASC (A$) + 800, ASC (B$
      ) + 128
13   PRINT "PRESS <N> TO END.";: GET
      S$: PRINT
14   IF S$ <  > "N" THEN  PRINT : GOTO
       10
20   POKE 56,0: POKE 57,3: REM  SE
      T RDKEY VECTOR
21   IF  PEEK (54) = 189 AND  PEEK
      (55) = 158 THEN  CALL 43089:
       GOTO 220: REM  SET RDKEY VE
      CTOR OF DOS
22   POKE 793,96
220   CALL 64352: POKE 1013,76: POKE
      1014,17: POKE 1015,3
230   VTAB 2: PRINT "JNEW KEYBOARD
       SUBROUTINE READY."
500   DATA  32,27,253,142,31,3,56,
      233,128,170,189,32,3,174,31,
      3,96,169,0,133,56,169,3,133,
      57,32,81,168,96,0
```

□

任意定義鍵盤

```
15  REM   **********************
20  REM   *                    *
25  REM   *  DEFINE KEY BOARD  *
30  REM   *                    *
35  REM   *  PRESS 'ESC' AND   *
40  REM   *  ANOTHER KEY TO    *
50  REM   *       QUIT.        *
60  REM   *                    *
65  REM   *  'CALL 768' AFTER  *
70  REM   *   PRESS RESET.     *
75  REM   *                    *
80  REM   **********************
85  REM
100  DIM X(95),Y(95): TEXT : HOME

110  FOR CHR = 224 TO 255: POKE 6
     29 + CHR,CHR: NEXT
120  PRINT "NORMAL KEYS --->"
130  VTAB 13: PRINT "----------
     ----------------------------
     "
140  VTAB 15: PRINT "CONTROL KEYS
      -->"
150  FOR O = O TO 95: READ X(O),Y
     (O):CHR = O: GOSUB 550: NEXT
     : POKE 34,22
200  HOME : PRINT "WHICH KEY YOU
     WANT TO CHANGE? ";: GOSUB 70
     0:O = CHR: IF O = 27 THEN  GET
     A$: IF A$ <  >  CHR$ (27) THEN
     300
210  GOSUB 600: PRINT "CHANGE TO
     WHAT";: GOSUB 700: GOSUB 600

220  GOSUB 550: IF O < 65 OR O >
     90 OR CHR < 65 OR CHR > 90 THEN
     250: REM   IF BOTH A APILHA-
     BET, CHANGE THE LOWER CASE.
230 O = O + 32:CHR = CHR + 32: GOSUB
     550: IF O = 104 OR O = 109 OR
     O = 117 THEN 250: REM      I
     F NOT RETURN , <- AND -> ALS
     O CHANGE THE CTRL KEYS.
240 O = O - 96:CHR = CHR - 96: GOSUB
     550
250  GOTO 200
300 A$ = "300:A9 OB 85 38 A9 3 85
      39 4C EA 3 20 1B FD A8 B9 9
     5 2 A4 24 60 N D823G": FOR I
      = 1 TO  LEN (A$): POKE 511 +
     I, ASC ( MID$ (A$,I,1)) + 12
     8: NEXT : POKE 72,0: CALL  -
     144
320  PRINT : PRINT "SAVE TO DISK
     ";: GET A$: PRINT A$: IF A$ =
     "N" THEN  CALL 768: GOTO 340

325  IF A$ <  > "Y" THEN 320

330  INPUT "FILENAME = ";NAME$: CALL
     768: PRINT  CHR$ (4)"BSAVE "
     NAME$",A $300,L $96": GOTO 3
     40
340  HOME : TEXT : END
540  REM
          PRINT VARIABLE CHR AND
     CHANGE IT FROM KEY BOARD

550  POKE 789 + O,CHR + 128: IF O
      > 95 THEN  RETURN
560  VTAB Y(O): HTAB X(O)
600  IF CHR > 32 THEN  PRINT " " CHR$
     (CHR)" ": GOTO 670
610  IF CHR = 8 THEN  PRINT "<- "
     : GOTO 670
620  IF CHR = 13 THEN  PRINT "RET
     ": GOTO 670
630  IF CHR = 21 THEN  PRINT "->
     ": GOTO 670
640  IF CHR = 27 THEN  PRINT "ESC
     ": GOTO 670
650  IF CHR = 32 THEN  PRINT "SPC
     ": GOTO 670
660  PRINT " ";: INVERSE : PRINT
      CHR$ (CHR + 64): NORMAL : PRINT
     " "
670  RETURN
690  REM   ALLOW CTRL-SHIFT-@ TO
     CHANGE
700  GET A$: IF A$ = "" THEN CHR =
     O: GOTO 720
710 CHR =  ASC (A$)
720  RETURN
810  DATA 31,16,5,19,18,21,12,21,
     11,19,10,17,14,19,17,19,35,8
     ,25,17,23,19,26,19,29,19,37,
     6,21,21,28,17
820  DATA 31,17,4,17,13,17,8,19,1
     6,17,38,8,15,21,7,17,9,21,19
     ,17,6,21,1,6,24,20,24,20,21,
     20,20,12
830  DATA 20,12,3,3,6,3,9,3,12,3,
     15,3,18,3,21,3,24,3,27,3,33,
     3,32,7,27,10,36,4,30,10,33,1
     0
840  DATA 30,4,3,4,6,4,9,4,12,4,1
     5,4,18,4,21,4,24,4,27,4,33,4
     ,32,8,27,9,36,3,30,9,33,9
850  DATA 31,5,5,8,18,10,12,10,11
     ,8,10,6,14,8,17,8,20,8,25,6,
     23,8,26,8,29,8,24,10,21,10,2
     8,6
860  DATA 31,6,4,6,13,6,8,8,16,6,
     22,6,15,10,7,6,9,10,19,6,6,1
     0,24,9,24,9,24,9,21,9,1,23
997  REM
998  REM   *** TUNG WAI YIP ***
999  REM    ***  2/6/1984 ***
```

任意定義鍵盤

W.T.Cheng

重新定義鍵盤的程式實在不少，大多數都是以機械碼編寫，用BASIC語言寫的也有，但是若這些程式未能配合監督程式，它們的應用是有限的，換言之，只有程式被執行時才有效，若要運行其他程式，它們便會失效，鍵盤沒有按用家要求而更改。

幸好APPLE的KSW 提供了一個途徑。每當電腦程式（無論BASIC或用監督程式中的輸入副程式的機械碼），要求輸入都會執行 JMP（$0038），此處原意是可讓其他周邊裝置（除了鍵盤）也可作輸入，所以平時$38，$39是載著鍵盤輸入程式（即從$FD1B開始）。我們只需把此兩個BYTES內容更改，指向用家程式，由用家將不同值存入累加器內，再跳離副程式（即RTS或$60），監督程式便會把累加器內容當作鍵盤輸入，達到用家改變鍵盤重新定義的目的。對於磁碟用戶，此等改變要告知DOS，DOS從$3EA 的程式會爲用戶建立KSW，即把$38與$39的內容存入DOS，否則，每次DOS要求輸入時，都會把舊有$38，$39的值（即$1B與$FD）存入$38，$39內，中斷用家的改變。

另外，當RESET鍵被按下後，監督程式會把$38 $39的值更新爲$1B與$FD，所以用家若要永遠使用，RESET VECTOR（位於$3F2的三個BYTE）便要更改，使它跳往更改$38，$39的內容，亦即重新接駁$38，$39要用家的程式。

附上的程式可以重新定義鍵盤，除了RETURN鍵外，所有鍵也可更改，當然可以全部改爲一個字碼，程式是利用查表方式，先輸入鍵碼，除去MSB（即BIT7），用作索引，取得該碼的「代以值」，存入累加器，再回到監督程式，從而改變了鍵碼的值。

程式執行時會先要求用家更改鍵盤代碼，若用家不想改或不再更改時，可按RETURN 脫離輸入更改碼，此時，鍵盤值會按用家所改而變更，用家若要保存更改了的代碼作日後用，可以存入DISK或TAPE ，需要時LOAD 入並執行這個程式便可以，起始地址，結尾地址與長度會在用家按RETURN 後印出，以供用家存取。最後程式佔用了兩個PAGE，所以程式會把鍵碼表，重置副程式，與輸入用的閱表程式移入$300 ，以免被BASIC的字串變量劃花程式，

```
*9400.95F0

9400- A9 00 8D F2 03 A9 03 8D
9408- F3 03 49 A5 8D F4 03 20
9410- 36 94 A9 16 85 38 A9 03
9418- 85 39 AD BF 9D C9 AD D0
9420- 04 18 4C D1 9D 4C 03 E0
9428- 86 06 20 1B FD 29 7F AA
9430- BD 24 03 A6 06 60 20 58
9438- FC A0 00 B9 52 95 F0 07
9440- 99 80 07 C8 4C 3B 94 20
9448- A1 94 88 C9 8D F0 5B AA
9450- C9 A0 90 05 99 80 07 B0
9458- 0D 18 69 40 48 A9 DE 99
9460- 80 07 68 99 81 07 A0 00
9468- B9 6E 95 F0 07 99 A8 07
9470- C8 4C 68 94 20 A1 94 88
9478- C9 A0 90 05 99 A8 07 B0
9480- 0F 48 18 69 40 48 A9 DE
9488- 99 A8 07 68 99 A9 07 68
9490- 48 8A 29 7F AA 68 9D D2
9498- 94 A9 00 20 A8 FC 4C 36
94A0- 94 AD 00 C0 10 FB 2C 10
94A8- C0 60 20 58 FC A0 00 B9
94B0- 7C 95 F0 07 20 ED FD C8
94B8- 4C AF 94 A0 24 B9 11 94
94C0- 99 FF 02 88 D0 F7 A0 80
94C8- B9 D1 94 99 23 03 88 D0
94D0- F7 60 80 81 82 83 84 85
94D8- 86 87 88 89 8A 8B 8C 8D
94E0- 8E 8F 90 91 92 93 94 95
94E8- 96 97 98 99 9A 9B 9C 9D
94F0- 9E 9F A0 A1 A2 A3 A4 A5
94F8- A6 A7 A8 A9 AA AB AC AD
9500- AE AF B0 B1 B2 B3 B4 B5
9508- B6 B7 B8 B9 BA BB BC BD
9510- BE BF C0 C1 C2 C3 C4 C5
9518- C6 C7 C8 C9 CA CB CC CD
9520- CE CF D0 D1 D2 D3 D4 D5
9528- D6 D7 D8 D9 DA DB DC DD
9530- DE DF E0 E1 E2 E3 E4 E5
9538- E6 E7 E8 E9 EA EB EC ED
9540- EE EF F0 F1 F2 F3 F4 F5
9548- F6 F7 F8 F9 FA FB FC FD
9550- FE FF C3 C8 C1 D2 C1 C3
9558- D4 C5 D2 A0 D4 CF A0 C2
9560- C5 A0 C3 C8 C1 CE C7 C5
9568- C4 A0 BA A0 60 00 C3 C8
9570- C1 CE C7 C5 A0 D4 CF A0
9578- BA A0 60 00 CB C5 D9 C2
9580- CF C1 D2 C4 A0 D2 C5 C4
9588- C5 C6 C9 CE C5 C4 8D C6
9590- CF D2 A0 C6 D5 D2 D4 C8
9598- C5 D2 A0 D5 D3 C5 AC 8D
95A0- D3 C1 D6 C5 A0 C9 D4 A0
95A8- CE CF D7 A1 8D D3 D4 C1
95B0- D2 D4 A0 C1 C4 C4 D2 C5
95B8- D3 D3 A0 BD A0 A4 B9 B4
95C0- B0 B0 8D C5 CE C4 A0 C1
95C8- C4 C4 D2 C5 D3 D3 A0 A0
95D0- A0 BD A0 A4 B9 B5 C6 B0
95D8- 8D CC C5 CE C7 D4 C8 A0
95E0- A0 A0 A0 A0 A0 A0 A0 BD
95E8- A0 A4 B0 B1 C6 B1 8D 8D
95F0- 00
```

任意定義鍵盤

其餘部份如印出指示和更改鍵碼的程式將會被消去，這樣可以減少記憶的使用量！

用家可以在監督程式下輸入目標碼（即機械碼）或用滙編器輸入此程式。

程式註釋如下：

LINES 9—14 ：

設置RESET VECTOR。

LINE 15 ：

跳至更改鍵碼程式。

LINES 16—19 ：

設置KSW。

LINES 20—25 ：

檢查是否存有DISK（程式至此終結）若否，跳至A—PPLESOFT BASIC暖起動，若有，把KSW存入DOS內，並用CLC來使DOS跳至BASIC暖起動（$9DD1的程式已包括兩個動作）。

LINES 26—32 ：

載存監督程式的輸入，亦即$38，$39所指向的地址。

注意，LINES16—32將會連同TABLE移往PAGE3。

LINES 33—85 ：

更改鍵碼的程式，用定會被要求更改鍵碼，並印出鍵碼給用家，CONTROL KEY 會被印出，如CTRL—C被印爲∧C。LINES33—55是將要被更改的字碼輸入LINES 56—76是改變爲那個字碼。LINES77—82是把更改的字碼ASCII碼存入表（TABLE）內。

LINES 86—89 ：

讀取鍵盤。

LINES 90—96 ：

印出程式起始，終結地址及長度。

LINES 97—107：

把部份控制程式存入$300內。

LINES 108—130：

字碼表。此表已包括小楷，若用家未有小楷，可以用其他字碼代替，但不可取消，不然，$94C7要改爲$60並從新設置TABLE1，TABLE2與TABLE3 的起始地址。□

噴墨PRINTER新突破

惠普公司最近推出配合私人電腦使用的新型HP 2225 型噴墨印字機，這種印字機操作寧靜（噪音低於50分貝），每部售價港幣三千九百五十元。

這種印字機可與惠普各種手提式或座枱式私人電腦配合使用，包括HP150觸幕式私人商用電腦。它還可以和其他牌子的私人電腦配合使用。

噴墨印字機是利用墨水通過印字頭中的微孔，噴在紙上形成字符。由於印字頭與其他類型的印字機不同，不會碰觸到紙上，所以噪音大大減低。

據惠普香港有限公司電腦總經理李國正表示：「噴墨印字技術雖然現時有了新的發展，但它一直是價格昂貴，補充墨水時很污髒及難於維修的。現在HP 2225印字機採用的新型噴墨機構則沒有墨水管道，不會堵塞，無需充注墨水池，也沒有活動零件，不會磨損。」

HP 2225 型印字機把印字頭和墨水池結合在一起，構成一個方便處置的單元。當機心中的墨水用完後，用戶可換上一個裝滿墨水和有新印字頭的新單元，這種單元的價格不超過港幣七十元。

HP 2225 型噴墨印字機的關鍵特點如下：

- ●印字速度每秒150 個字符；
- ●11×12點矩陣字符；
- ●印字時噪音級低於50分貝；
- ●多種印字呎吋；
- ●印黑體和重點綫亦不會減慢印字速度；
- ●有十一種其他語言支援；
- ●採用8½×11英吋的紙度，既可用單張，也可用叠折紙張；也可以用21.0×29.7 厘米 A4 呎吋 的紙張；
- ●印字機的呎吋是11.5 英吋寬，8.1英吋長，3.5英吋高；
- ●重量5.5 到6 磅；
- ●墨水筒的壽命平均是500頁。
- ●備有 CENTRONIX ，HP—IB和HP—IL(電池供電) 介面；
- ●電池壽命平均200頁（僅適用於HP—IL 介面）；
- ●索引式和摩擦式進紙；
- ●全面的繪圖能力（每英吋192×96點或96×96點）。

這種印字機可以使用任何紙張，但以使用噴墨紙效果最佳，噴墨紙的價格和重量與書紙近似。□

捉蟲訓練——匿藏起來的蟲

今期〔捉蟲訓練〕的題材是來自一位讀者的投稿——匿藏起來的蟲。

程式中的蟲也有很多種，有些會產生語法的錯誤，有些在執行時看得到，有些却是非常難找。執行時很正常，但也會有蟲的。

附列的程式共可分爲五部份：

一、要求用家輸入亂數函數用的數子，
二、利用種子產生一陣列的亂數，
三、印出亂數陣列，
四、用浮泡法去把陣列排序，
五、把已分類的陣列印出。

程式看起來很好，沒有甚麼不對，執行起來很美觀，排序似乎沒有錯。然而，在這程式中却有蟲，使得程式未能達到上述的所有目的，聰明的讀者，試把這條蟲找出來吧！□

```
10 N = 10
30  TEXT : HOME : VTAB 10: INPUT
    "PLEASE ENTER THE SEED FOR T
    HE RANDOM FUNCTION : ";S
40  FOR I = 1 TO N
50 A(I) =  RND (S)
60  NEXT I
63  HOME
65  VTAB 4: PRINT "UNSORTED ARRAY
    :": FOR I = 1 TO N: PRINT A
    (I): NEXT
70  FOR I = 1 TO N - 1
80  FOR J = N - 1 TO I STEP  - 1
90  IF A(J) > A(J + 1) THEN Z = A
    (J):A(J) = A(J + 1):A(J + 1)
     = Z
100  NEXT J: NEXT I
105  POKE 32,20: VTAB 4: PRINT "S
    TORED ARRAY :": FOR I = 1 TO
    N: PRINT A(I): NEXT
110  POKE 32,0
```

聰明的讀者請將 正確的答案寄來給我們。我們將會送出一張程式磁碟給最佳答案的一位讀者。答案將在下期揭曉。

請連同姓名、地址、聯絡電話寄來九龍中央郵政信箱 71193 號。信封面請註明「捉蟲」。

同時，我們極之歡迎讀者們自己設計一些類似的捉蟲訓練題目投寄給我們。程式長度沒有限制，當然越精簡越好，程式以 APPLESOFT 編寫，並需符合下列各項要求。

1、程式必須是 APPLESOFT ，任何 POKE 機械語言的子程式或 CALL 該子程式必須正確。
2、程式必須能出現有某些結果，而不會出岔子，換言之絕不能有 SYN—TAX ERROR出現。
3、程式必須附加簡短說明。
4、必須連同答案寄來。

假如閣下投寄來的捉蟲訓練程式是有水準，我們將贈閱十二期雜誌給你，或者再多一年的免費讀訂。

投稿請在信封面寫上「投稿捉蟲」。□

多餘的副程式——答案

今次參加〔多餘的副程式〕捉蟲的朋友可能是心急了一點，許多答案都漏了少許，例如〔:〕，令到程式不能執行。我們相信這是無心之失，並非是讀者水準不足。希望讀者們留意在下次的投稿中繕寫清楚答案及說明。另一樣我們需要再重覆的在取捨最佳答案時，解釋說明程式所存在的〔蟲〕是十分重要的，有些讀者依然只是將改正後的答案程式寄來，而不附加任何解釋，無論程式如何優良，也必不能入選，我們舉辦這欄的目的，是希望讀者能藉此機會，訓練加強 APPLESOFT 及設計程式原理的基礎，所以詳盡而有邏輯的附加說明是必須的。

上期得獎者是住在秀茂坪的梁永建讀友。他的答案相當簡潔而說明解釋亦十分清楚。

另一位較爲優異的是住在土瓜灣的張錦文讀友。可惜他的附加說明未及梁永建清晰，故未能得獎。希望張錦文和其他的讀者能再接再勵，參加今期的捉蟲。□

得獎者梁永建

原來的程式，有兩個主要部份，一個是劃出不同顏色，由大到小的正方形，而另一部份就劃出黑色的由小到大的正方形，只要將以上的兩部份合併，便可以省去多餘的副程式。

首先，設定X是由0至39的變數，再設定Y爲39—X，這樣Y就變成一個39至0的變數，在Y＝0至19的部份，X和Y的距離是在減小的，所以正方形的邊也在減小，一個個由大到小的正方形便可在迴路中得到，而在X＝20 TO 39 的時候，X和Y的距離是不斷地增加，正方形的邊也在增大，所以由小到大的正方形也可得到。這樣便可以建立一個由大到小，又由小變大的正方形在一個迴路裡。

但是我們知道兩部份所劃的顏色是不同的，只要加上 CO—LOR＝RND（1）＊ 16 ＊（X＜ 20）這一句，假如X小於20，（X ＜ 20）的值是一，不影響隨機顏色的設定，當→

讀者園地

X大或等於20，（X＜20）的值是零，所以 COLOR 被設爲 0。這樣將副程式，放入迴路中，便可以省了這個多餘的副程式。

以下是更改過的程式：

```
10  GR
20  FOR X = 0 TO 39
30  COLOR= 0
40  IF X < 19 THEN  COLOR=  RND (
     1) * 16
50  HLIN X,39 - X AT X
60  HLIN X,39 - X AT 39 - X
70  VLIN X,39 - X AT X
80  VLIN X,39 - X AT 39 - X
90  NEXT X
100  GOTO 20
```

優異答案張錦文

第七期的捉蟲訓練——多餘的副程式一文中，其程式是沒有BUG 的，而且也能執行，只要長了一點兒罷了。

其實在 BASIC 述句 VLIN e1，e2 AT e3中，e2 不一定要大於 e1，若e1 大於 e2 則會繪劃e2 至e1部份。故此，程式中大可用 FOR X＝0 TO 39，而無須分開兩段，當然也不再需要副程式了。至於顏色方面，只須測試X是否小於20來決定用黑色或彩色來繪劃。改良後程式較爲簡短，但是却犧牲了程式的明讀性了。改過的程式如下：

```
10  GR
20 Y = 40
30  FOR X = 0 TO 39
40  COLOR= 0
50  IF X < 20 THEN  COLOR=  RND (
     1) * 16
60 Y = Y - 1
70  HLIN X,Y AT X
80  HLIN X,Y AT Y
90  VLIN X,Y AT X
100  VLIN X,Y AT Y
110  NEXT
120  GOTO 20
```

30.BFFFR 拆帶方法

小弟今次是初次投稿，是有關於30 BFFFR 帶的拆法，此方法小弟亦曾嘗試及多次成功，爲了與電腦時代之讀者分享我的成果，特此投稿。

「30.BFFFR 是最近流行的保護方法，它涵蓋了第 0 頁至第三頁的敏感位址，而且是一段帶，若將它存至自由位址區，便會得到 ERROR 的訊息（因帶長的關係），考慮此種保護方法於讀到最後位址 BFFF 時，會轉由 CO—UT 字元輸出副程式控制，故此避開 30.3F 便可進行拆帶工作，首先將原本30.BFFF 的位址由40開始錄起至 BFFF 止，你可能奇怪這段帶豈不是有 16 BITS位址，沒有錄齊麼？是的，所以這個拆帶方法也有它的缺點，若程式本身佔用位址（有用位址）至BFFF 時，則程式執行時會發生錯誤，但佔用至 BFFF 的程式極小，各位讀者大可不必担心。錄完之後， ERROR 會出現，告訴我們程式並未錄齊，CO—NTROL—RESET後， CALL—151進入MONITOR PROGRAM，（RESET後，會像初開機時一樣），首先將1010.BFFF 錄了下來，因 CONTROL—RESET 之後，0.800 之間位址已變（因 COLD RESET—CYCLE之故），因而不能一次過 CO—PY ，要分兩段，隨之我們再將原帶由1030錄至2000，當然另一次 ER—ROR 會出現，然後將此段位址也錄下來，這時，我們只要利用程式呼叫 1000.BFEFR，再呼叫 30.1000R 便可。

讀者：莫信

道歉

上期空戰驚魂程式中行號180 有多少許錯誤，現修訂如下：

```
180 ZE = ZE + 1: IF ZE = 52 THEN
     ZE = 1:.RESTORE : FOR J = 1 TO
     14: READ D: NEXT
```


〔VOCABULARY 1-2-3〕

教育軟件

電腦時代讀服務部設有〔VOCABULARY 1-2-3〕軟件的示範，讀者若想深入了解這套軟件，可 到服務部參觀。

讀者園地

讀者服務部通訊

＊〔特惠優待讀者證〕經已於六月底前全部寄出，相信有來信申請的讀者經已收到。由於申請信之中有相當數量是未附有回郵信封，這批申請目前尚在處理中，相信要一段時間方能完成。希望這批讀者不要心急，同時，敬請未附同回郵信封的讀者補寄一個，這會加快我們處理的速度。至於在六月十五後寄來申請表格的讀者，我們留待第二次的公開接受申請時才處理。

□ □

＊持有優待咭的讀者在我們服務部購物可獲得特惠的優待，希望你們能好好利用此咭。

□ □

＊由八月開始，我們將定期發行一份〔電腦時代通訊〕，內容主要是報導服務部最新的動態消息，特價貨名單及價錢，及讀者之間的消息經驗交流，〔電腦時代通訊〕是一份不牟利通訊錄，免費寄給持有〔特惠讀者優待證〕的讀者，短期內不會公開發行。未有來信登記的持證讀者，請盡早寄來通知。

□ □

＊第七期的程式磁碟由於 HELLO PROGRAM 過長（因爲要製造 BOOT 機時發聲效果，佔用去不少記憶空間，故磁碟上部份有一定長度的 PROGRA—M在 RUN 過 HELLO 程式後，是沒有足夠記憶放置，而且會出現〔PR—OGRAM TOO LONG〕或重覆返回 MENU 的情況。這並非顯示磁碟本身或其中的程式有問題。解決辦法：在 BOOT 機時，按下 CONTROL C，使電腦只載入 DOS，而不 RUN HELLO，（螢幕會有 BREAK字眼出現），然後打入 CATALOG 將磁碟之 FILE 顯示，便可隨意 RUN 任何程式了。

〔軟件自由談〕

公開徵稿

本刊由第八期開始增加一欄〔軟件自由談〕公開園地，歡迎讀者自由投稿，發表個人對各種軟件（無論商用、應用、遊戲等等）的意見、使用心得及評價，好讓他人能分享及增加廣泛認識，來稿字數長短不限，一經刊登，即有稿酬。投稿辦法可參閱〔歡迎投稿〕。

＊＊服務部於 8 月初開辦 PRODOS 使用法研討講座，討論如何使用 PRODO—S，每次只收最多六至七人，名額有限，有關詳情可來電查詢或親臨服務部報名。

□ □

＊＊ COMPUHOUSE

每期製作電腦時代程式磁帶推出爲沒有磁碟機的讀者帶來喜訊。程式磁帶每盒特價十元。可於指定的特約經銷處購買或親臨服務部購買。

□ □

＊＊COMPUHOUSE 計劃短期內推出 PRODOS版本程式磁碟，敬請讀者留意。

□ □

由於今期稿擠，PASCAL專欄暫停一期。

□ □

＊ DENNIS WONG 讀者來信詢問 LOCK—IT—UP 程式的使用方法，讀者們可否嘗試投稿講述 LOCK—IT—UP 的使用。

□ □

＊ 馮兆民讀友來信詢問及如何鍵入所登出的機械語言程式。有關如何鍵入機械語言及 SAVE，可參閱本刊第三期〔機械語言程式輸入方法〕一文。

各期程式磁碟磁帶

第7期程式磁碟

（1）會說話的蘋果
（2）私人財政預算系統
（3）私人收入會計系統
（4）高解像字符繪劃器
（5）圖像轉載器
（6）螢幕文本編輯處理器
（7）濃縮圖片程式修訂加強版本
（8）蘋果杯大賽車
（9）十六個難度的食鬼遊戲
（10）高爾夫球遊戲。
（11）空戰驚魂
（12）美麗圖案設計家
（13）更多色彩
（14）列印的困擾
（15）十一條空 SECTOR
（16）捉蟲訓練
（17）80字行咭例子程式

第7,8期程式磁帶

編號：C#7

（一）十六種難度的食鬼遊戲
（二）更多色彩
（三）空戰驚魂
（四）捉虫問題
（五）捉虫答案

編號：C# 8

（一）最簡單的拆帶程式
（二）英文打字練習
（三）原子黑洞
（四）蛇之噩夢
（五）收銀機
（六）任意鍵盤（三個）

各期程式磁碟磁帶總目錄

1

1 突破傳統的鍵盤
2 捉球遊戲
3 勇闖電腦室
4 迷宮救美
5 將 APPLESOFT 變爲 FORTRAN
6 上、下、左、右移動文字畫面
7 測試工具— BOOLEAN 程式
8 騰空更多磁碟空間一搬移HELLO
9 重現刪除之檔案
10 電腦時代月曆
11 COMPUTEXT 文字處理
12 時日相距計算程式
13 增強英文字彙能力
14 幼童數學四則運算練習
15 過四關

2

1 貨櫃處理
2 沙蟹幻方局
3 巧破密令
4 美術字繪劃器
5 小階字母處理
6 程式轉載器
7 彩色電視／監視器色彩測試
8 保留遊戲積分記錄
9 數學套件程式
10 磁碟修護工具
11 加送電腦時代月曆圖像一個

3

1 DOS 的自動同步追踪器
2 動量不滅定律模擬法
3 自製磁碟索引表封套
4 蘋果編曲器
5 幼童認數字初階
6 賽跑遊戲
7 步步追踪
8 DOS KILLER
9 圍棋遊戲
10 巴哈的古典樂章名作
11 旋轉螢幕畫面
12 雜誌目錄分類歸檔系統
13 神經病醫生—神經緊弛體操

4

(1) DOS 操作指令簡化器
(2) TEXT FILE 有些甚麼內容？
(3) 立體迷宮
(4) 英文串字練習
(5) 如何以機械語言來修補 DOS
(6) 電腦動畫製作原理。
(7) 豐富設計的學前兒童教育程式
(8) 應用數學套件程式(二)
(9) DOS WRITER
(10) 穿梭機哥倫比亞號機
(11) 找回失落了的 STEP 及 TRACE
(12) TRACER控制常式
(13) 將 RAM 卡作爲磁碟機使用
(14) 黑房計時器
(15) 太陽系模擬系統

5

1 私人約會備忘日曆
2 三度空間顯像系統
3 將 CP／M 檔案轉爲 DOS
4 蘋果萬花筒
5 高解像度文字產生器
6 修訂 DOS 操作指令簡化器
7 濃縮圖片儲存空間
8 幼童數學練習—— 數字咭
9 加數機大登場
10 備忘 MEMO紙
11 光速計數器
12 剩餘磁碟空 SECTOR數目
13 16進制輸入數字鍵盤
14 問題棋盤
15 神偷智破夾萬
16 隔山打牛
17 大空船降落任務

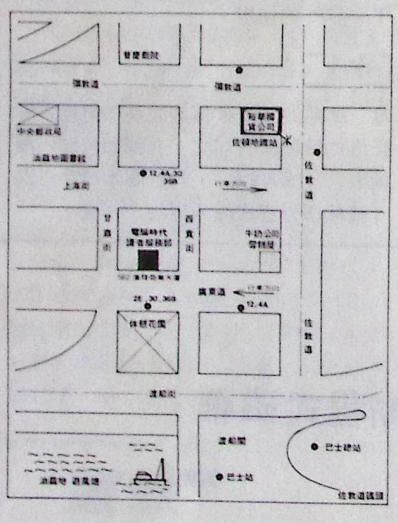
6

(1) 私人會計系統。
(2) APPLESOFT 擴展版本。
(3) GO }〔，一個取代 HELLO 的程式
(4) APPLESOFT 偵錯系統。
(5) DOS 指令再修訂版本。
(6) 布倫時鐘。
(7) AMPERSAND 數基轉換器。
(8) 完整索引錄。
(9) 砌積木。
(10) 神奇數字方塊。
(11) 亡命飛車。
(12) 迷宮搜秘。
(13) 太空怪侵略戰。
(14) 累加遊戲。
(15) 爲磁碟命名。
(16) 任意定義鍵盤。
(17) 捉蟲訓練問題。

精選程式磁碟

編號：CA00A
內容：遊戲及教育程式
遊戲程式
1、捉球遊戲（第一期）
2、勇闖電腦室（第一期）
3、迷宮救美（第一期）
4、過四關（第一期）
5、沙蟹幻方局（第二期）
6、貨櫃處理（第二期）
7、賽跑遊戲（第三期）
8、步步追踪（第三期）
9、圍棋遊戲（第三期）
10、立體迷宮（第四期）
11、穿梭機哥倫比亞號（第四期）
教育程式
12、增強英文匯能力（第一期）
13、幼童數學四則運算（第一期）
14、數學套件程式（第二期）
15、動量不滅定律模擬法（第三期）
16、幼童認數字初階（第三期）
17、豐富設計的學前兒童教育程式（第四期）
18、應用數學套件程式之二（第四期）

編號：CA00B
內容：應用程式精選
1、電腦時代月曆（第一期）
2、COMPUTEXT 文字處理（第一期）
3、美術學繪劃器（第二期）
4、彩色電視／監視器色彩測試（第二期）
5、保留遊戲積分紀錄（第三期）
6、自製磁碟索引封套（第三期）
7、APPLE 編曲器（第三期）
8、雜誌目錄分類歸檔系統（第三期）
9、DOS WRITER（第四期）
10、電腦動畫製作原理（第四期）
11、將 RAM 咭作爲磁碟機使用（第四期）



精選程式磁碟及磁帶

讀者服務部由即日開始推出兩張全新製作的程式精選磁碟：一張是「遊戲及教育程式」，另一張是「應用程式」，兩者均是將電腦時代第一至第四期內精彩的程式融合爲一。每張訂價港幣 45 元。

與此同時，亦推出各期遊戲程式的磁帶版本，每盒訂價港幣十元。喜歡玩遊戲磁帶的讀者記著不要錯過。精選程式磁碟及第8期程式磁帶可在下列指定地點購買：

(1) 百老滙冲印公司 第一店
九龍旺角彌敦道 758 A地下（近凱聲戲院）。電話：3—816931

(2) 百老滙冲印公司 第十四店
香港銅鑼灣軒尼斯道 529號（三越百貨公司對面）電話：5—770181

或親臨電腦時代讀者服務部購買
查詢電話3—312007。

各期程式磁帶

編號：C#1
1、捉球遊戲
2、勇闖電腦室
3、迷宮救美
4、過四關
5、增強英文字彙能力
6、幼童數學四則運算
7、時日相距計算程式

編號：C#2
1、沙蟹幻方局
2、貨櫃處理
3、數學套件程式
4、巧破密令
5、彩色電視／監視器色彩測試

編號：C#3
1、賽跑遊戲
2、步步追踪
3、圍棋遊戲
4、動量不變定律模擬方法
5、幼童認數字初階

編號：C#4
1、豐富設計的學前兒童教育程式
2、應用數學套件程式之二
3、立體迷宮
4、穿梭機哥倫比亞號
5、英文串字練習

編號：C#5
1、加數機大登場
2、神偷智破夾萬
3、隔山打牛
4、太空船降落任務
5、蘋果萬花筒
6、幼童數學練習 ——數字咭

編號：C#6
1、累加遊戲
2、亡命飛車
3、迷宮搜秘
4、太空侵略戰
5、切積木
6、神奇數學方塊

歡迎補購各期程式磁碟磁帶或雜誌

請前往 電腦時代讀者服務部/COMPUHOUSE

佐敦廣東道582號廣發商業大厦八樓三室。TEL :3-312007

領先 Apex 助你掌握電腦！

私人及小組電腦証書課程

1. Programmer's Full Cert.Training Course
2. Computer Operation & Basic Programming
3. Advanced Basic
4. Disk Operating System (File Handling)
5. Wordstar and CP/M Operating System
6. dBase II and Utility Packages
7. Assembly Language, Cobol,Logo, Fortran or Pascal
8. Peachtree Inventory Control System
9. Peachtree Account System
10. 電腦磁碟折銷，保護及原理
11. 初級電腦裝嵌及修理
12. 初級兒童電腦操作及認識
13. 中文電腦操作，會考電腦補習

私人教授，天天開課。小組學習，興趣濃厚

領先私人電腦訓練中心　Apex Computer Training Centre

＊歡迎報名或電話查詢＊

地址：九龍深水埗青山道135號3樓11室　（近九江街，金盟商場對面，電梯2字）

電話：3-7280889，3-7299187

電腦公司及製造商請注意！

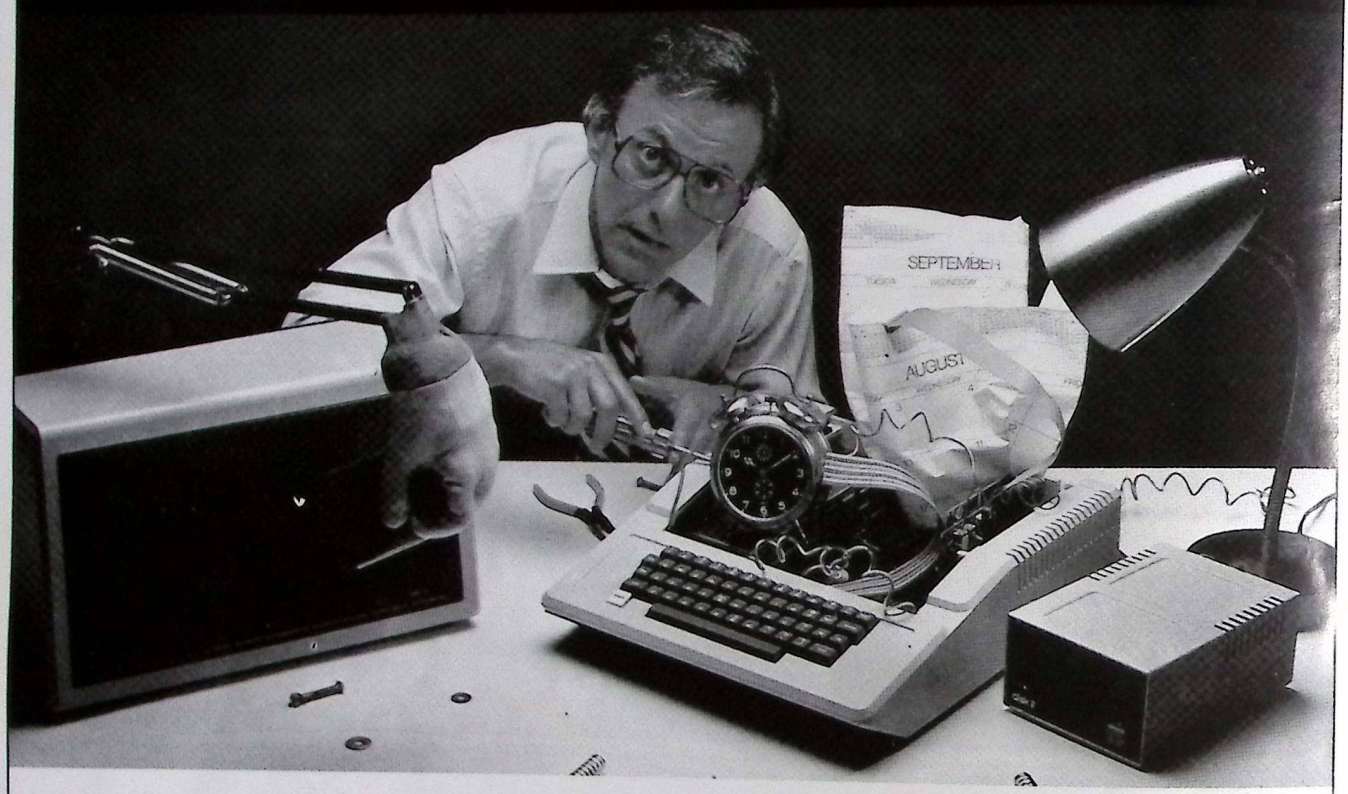
貴　公司是否希望將最新的電腦產品訊息傳遞給電腦用家知曉？那麼爲何不選擇一本擁有高消費能力兼廣泛層面讀者基礎的雜誌。

電腦時代就是貴　公司最適當的選擇。我們擁有廣泛層面的讀者。他們不但擁有高消費能力，亦是標準的微電腦用家。他們渴望知道貴　公司所出品或代理的最新產品訊息。敬請立即與本刊廣告部接洽。

電腦時代廣告部電話3-312007

何必自己動手

免費 由專業人員上門爲你修理微型電腦



請即加入本公司之保養及免費維修服務

24小時熱綫電話服務：3-306839

微型電腦保養維修中心

ROM & RAM computer ltd

SHOP: 582–592, Nathan Road, Basement Shop No. B35 Sino Centre, Kowloon. Tel: 3-306839
OFFICE: Rm. 1905 Witty Coml. Bldg., 1A–1J, Tung Choi St., Kowloon. TEL: 3-848129
Telex: 57551 ROMHX

COMPUTING AGE

電腦時代

第八期
一九八四年七月十五日出版
本刊每月十五日出版
每本訂價港幣八元正

編者註：

今期，我們新增加一欄〔軟件縱橫談〕主要是介紹及探討時下一些我們認爲有價值的軟件給讀者認識。目前本港的軟件市場可謂花樣繁多，琳瑯滿目，讀者們未必有時間去了解最新的軟件市場動態。我們希望透過此欄，幫助各位讀者。第一次要介紹的軟件是一個工具軟件〔DOUBLE TAKE〕。此軟件之功能出乎我們意料之好，對於日常使用電腦，它能提供許多方便及快捷效率。

在硬件方面，我們將會連續兩期討論Z80咭，讓大家能更深入了解認識這張週邊咭的功能。

軟件方面，有兩個值得讀者們留意的，其一是〔眞正要動腦筋的小學四則運算〕及〔用GRAPHICS MAGICIAN 編寫的遊戲—七海霸王DEPTH CHARGE〕。前者是供小童學習四則計算，程式完全按照日常課堂上的算術方法模擬而編寫，肯定對學童練習算術有極強的吸引力。而〔七海霸王〕主要是引導讀者們如何自己編寫有商業水準的遊戲軟件，作者麥法基詳細解釋了編寫遊戲的過程及技巧。

出　　版/ 電腦時代出版社
地　　址/ 廣東道582號廣發商業大厦703室
編　　輯/ 電腦時代編輯部
通訊地址/ 九龍中央郵箱71193號
通訊電話/ 3-312007
軟件支援/ CompuHouse
讀者服務部/ **廣東道582號廣發商業大厦八樓三室**
承　　印/ 永恒印刷公司
　　葵涌工業街美安工業大厦12字樓A座
發　　行/ 同德書局
　　九龍西洋菜街193號

今期內容

□ Program saved on disk

電腦人・電腦事

澳洲電腦多能智多星 37
噴墨印字機新突破 86
展覽會採用「輕觸電腦」作入場登記 56

應用軟件庫

□最簡單的拆帶程式 13
□打字練習 15
□APPLESOFT滙編器 17
□編印單個 LABEL 51

專欄

CP/M 專欄——隨機命令及工具程式 42
PRODOS縱橫——磁碟格式 46
□奇妙的 & 專欄 47
DOS3.3 專欄——用機械語言直接控制 DOS 指令 49

特稿

Z80咭的認識 7
將INTEGER轉爲APPLESOFT 34
RESET鍵的特殊問題 77

蘋果工具箱

□高解像頁處理工具箱 52
□簡易快捷的記憶移動方法 55

兒童電腦輔助教育

□眞正要動腦筋的小學四則運算 25

電子歡樂圈

□DEPTH CHARGE（七海霸王）（上） 57
□原子黑洞 62
□蛇之噩夢 67

軟件縱橫談

有廿四種功能的DOUBLE TAKE 38
60元一個COMMODORE MOUSE??? 41

學校電腦輔助教育

□自動三角函數曲線繪劃系統 68

精品店

□兩欄索引顯示 11
□收銀機 76
□任意定義鍵盤程式公開發表 83

讀者園地

□捉鼻訓練問題— 匿藏起來的蟲 87
更快的計數器 80
□改頁的閃動訊息 66
30. BFFFR 拆帶方法 88
訂閱 10 投稿需知 53 讀者通訊 89
第1至第7期程式磁碟總目錄 90

電腦時代程式磁碟

每張只售HK$29.90

電腦時代每期都刊登多個精彩的程式，其中必然有你喜歡的。當你讀完文章興緻勃勃的坐下來欲進一步深入了解程式的精彩處時，看見那大堆冗長的列表，就甚麼興趣也沒有了！那麼，你爲何不購買一張「電腦時代程式磁碟」呢！

每期出版的電腦時代所登載的程式，均全部紀錄入一張高品質的磁碟上，並隨雜誌同時出版。爲省卻打入程式的麻煩，徒然浪費你寶貴的時間，免除了花精神捉錯打字的痛苦，今日就請就近的特約經銷處或電腦時代讀者服務部/COMPUHOUSE購備一張。

程式磁碟每張只售港幣 29.90 元。

第8期程式磁碟內容：

*All Programs are in Apple version

（一）APPLESOFT滙編器
（二）最簡單的拆帶程式
（三）英文打字練習
（四）編印單個LABEL
（五）高解像頁處理工具箱
（六）簡易快速的記憶移動方法
（七）眞正要動腦筋的小學四則運算
（八）七海霸王（DEPTH CHARGE）
（九）原子黑洞
（十）蛇之噩夢
（十一）收銀機
（十二）改良的閃動訊息
（十三）兩欄索引顯示
（十四）自動三角函數曲綫繪劃系統
（十五）任意鍵盤（三個）
（十六）每期捉蟲問題
（十七）奇妙的 &

爲方便各區讀者購買程式磁碟，我們新增加多個特約經銷處：

以下爲各特約經銷處地址：

中環

百老滙攝影器材有限公司（第五店）
域多利皇后街3號（中環街市側）
電話：5－235513

亞當影音
士丹利街 28 A（陸羽茶樓側）
電話：5－240723

灣仔

金鐘廊書報店
金鐘廊第一、二店（美國餐廳對面）
（地鐵金鐘站）電話：5－292843

銅鑼灣

百老滙攝影器材有限公司（第十四店）
軒尼斯道529號地下（三越百貨公司對面）
電話：5－770181

太古城

百老滙攝影器材有限公司（第七店）
太古城中心第 232店
電話：5－672006

西環

創業雜誌中心
創業商場中心G 91 號（假日城遊戲機中心對面）
電話：5－478338

旺角

百老滙攝影器材有限公司 （第1,3,9,10店）
彌敦道758號在下（凱聲戲院側）
電話：3－816931
彌敦道746號地下（龍聲戲院側）
電話：3－943827
彼油街 26 號地下（近商務書局）
電話：3－884331
亞皆老街 60B 地下（世界書局對面）
電話：3－956367

青年圖書有限公司
亞皆老街 80 號昌明大廈H座1／F
（飛龍快餐店樓上）電話：3－918965

尖沙咀

智源書局
金巴利道 27－33號永利大厦2字樓A座
電話：3 － 678482 3－678414

荔枝角

百老滙攝影器材有限公司（第二店）
美孚新邨第四期百老滙街 79號（影都戲院側）
電話：3－7424923

觀塘

百老滙攝影器材有限公司（第四、十二店）
物華街3號地下（皇上皇側）
電話：3－419513
同仁街 35 號地下（大家樂側）
電話：3－449940

荃灣

百老滙攝影器材有限公司（第十一店）
沙咀道 230 A（戴麟趾夫人健康院對面）
電話：0－438926

深水埗

百老滙攝影器材有限公司（第十三店）
青山道148 號地下（華聲戲院對面）
電話：3－7295938

電腦時代讀者服務部/ CompuHouse

地址：九龍廣東道588 號廣發商業大厦 八樓三室
（近佐敦道渡船閘） 電話：3-312007

補購各期程式磁碟可前往讀者服務部／COMPU HOUSE

Apple is the trademark of Apple Computer Inc.